(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 October 2003 (23.10.2003)

PCT

(10) International Publication Number
WO 03/088125 A2

(51) International Patent Classification⁷: G06F 19/00

(21) International Application Number: PCT/US03/10961

(22) International Filing Date: 10 April 2003 (10.04.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/371,643 10 April 2002 (10.04.2002) US
60/371,643 10 April 2002 (10.04.2002) US
60/371,956 11 April 2002 (11.04.2002) US
60/371,871 11 April 2002 (11.04.2002) US

(71) Applicant (*for all designated States except US*): **TRANSTECH PHARMA, INC.** [US/US]; 4170 Mendenhall Oaks Parkway, Suite 110, High Point, NC 27265 (US).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): **SAWAFTA, Reyad, I.** [US/US]; 5506 Gate Post Court, Greensboro, NC 27455 (US). **BAUDRY, Jerome** [FR/US]; 3305 Valerie Drive, Champaign, IL 61822 (US). **KUTZ, Michael, E.** [US/US]; 103 Peach Orchard Drive, Greensboro, NC 27455 (US). **SUBRAMANIAN, Govindan** [IN/US]; 1835 Morgans Mill Way, High Point, NC 27265 (US).

(74) Agents: **CALKINS, Charles, W.** et al.; Kilpatrick Stockton LLP, 1001 West Fourth Street, Winston-Salem, NC 27101 (US).

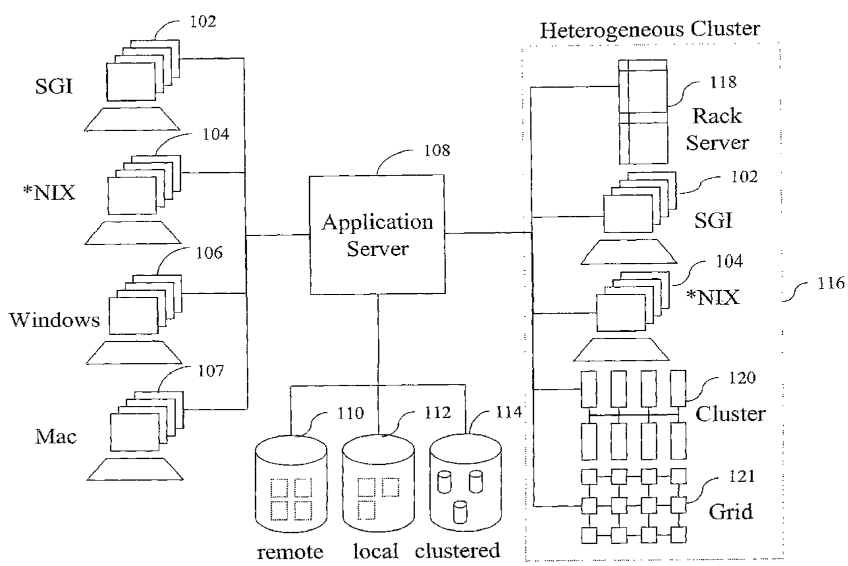
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) **Title:** SYSTEM AND METHOD FOR INTEGRATED COMPUTER-AIDED MOLECULAR DISCOVERY



(57) **Abstract:** Systems and methods for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications are described. In one embodiment, an integrated user interface provides the user with access to the capabilities from a broad array of commercial and custom application modules, such as calculation engines. In another embodiment of this invention a heterogeneous cluster provides the ability to divide the processing required for a molecular discovery process to increase efficiency and reduce the time required to perform molecular discovery.

WO 03/088125 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR INTEGRATED COMPUTER-AIDED MOLECULAR DISCOVERY

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority under 35 USC 119 from US provisional application serial number 60/371,644, entitled "System and Method for Data Analysis, Manipulation and Visualization", filed April 10, 2002; US provisional application serial number 60/371,956, entitled "System and Method for Data Analysis, Manipulation and Visualization", filed April 11, 2002; US provisional application serial number 60/371,643, entitled "System and Method for Integrated Computer-Aided Molecular Discovery," filed April 10, 2002; and US provisional application serial number 60/371,871, entitled "System and Method for Integrated Computer-Aided Molecular Discovery," filed April 11, 2002. The disclosure of each of these provisional applications is hereby incorporated herein by reference. This application also relates to US patent application serial number 10/120,278 entitled "Probes, Systems, and Methods for Drug Discovery," filed April 10, 2002 which is incorporated herein by reference. This application also relates to attorney docket number 41305-283186, filed simultaneously, entitled "System and Method for Data Analysis, Manipulation and Visualization" which is incorporated herein by reference.

NOTICE OF COPYRIGHT PROTECTION

FIELD OF THE INVENTION

This invention generally relates to computer-aided molecular discovery. This invention more particularly relates to a system and method for integrating diverse commercial and custom computer application modules to perform molecular discovery.

BACKGROUND OF THE INVENTION

Combinatorial chemistry historically required the screening of thousands of compounds to identify a lead compound or scaffold(s) for detailed analysis. The introduction of computational methods for identifying the scaffold(s) has increased the efficiency of the

screening by reducing the number of compounds to be analyzed. However, although the number of compounds to be analyzed has decreased, the number remains large.

Also, performing computer-aided molecular discovery has conventionally required the use of multiple incompatible systems or the use of a single application, which includes some but not all of the functionality required to efficiently and effectively identify compounds of interest. An application from one software vendor may include one module or a suite of modules for performing various tasks in the discovery process. Generally, the application modules from one vendor work together by design; however, they are generally incompatible with modules from other vendors.

Conventionally, chemists and biologists have addressed the incompatibility or ineffectiveness of existing systems in various ways. One approach has been to perform each step in the process using the best tool, regardless of what application it resides in. Following this approach is relatively slow and labor-intensive. The scientist must wait for each step to finish before beginning the next step. In addition, various modules from different applications are often incompatible, forcing the scientist to make changes to the input and output files manually in order to continue the process.

Because of the effort involved in attempting to integrate multiple incompatible systems, many scientists simply use a single application. Unfortunately, a single application may not perform every function or may not perform every function in an optimal manner. By following this approach, the scientist is forced to accept the shortcomings of a particular module within an application in order to avoid the manual processes involved in using the best modules of several applications.

In addition to the problems experienced by scientists, the computer and network administrators face multiple challenges in supporting these scientific applications. One challenge is to ensure that all applications, including the user interfaces, operate on the scientists' workstations. Another major challenge is ensuring that the performance of the system is acceptable. Some of these applications are computationally intensive and place great demands on the system on which they are installed.

The problem of ensuring adequate performance of a computationally-intensive application is conventionally addressed in a number of ways. The simplest approach is to buy more powerful servers. However, very powerful servers, such as supercomputers, are also very expensive. Also, one server may not be able to run all the diverse applications required by the scientists.

Other approaches for addressing these needs include implementing a grid computing architecture. One such approach used by some organizations outside the molecular discovery area is to use the unused computer cycles of computers on a network. For example, the Search for Extraterrestrial Intelligence (SETI) Institute (SETI.org) has established a program called SETI@Home (http://www.seti.org/science/setiathome.html), whereby individuals download a software module that allows SETI to use the unused computing cycles on the individuals' computers. However, as SETI states, "the data processing does not occur 'real time' so that interesting signals must be followed up at a later date". The SETI system does not attempt to determine which systems are available. When an individual's computer is not busy, for example, when the screensaver appears, the computer requests a "work unit" from SETI. If and when processing is complete, the computer transfers the file back to SETI. Ensuring that work is performed on a timely basis is very difficult with the SETI implementationand determining the status of a particular job is virtually impossible until that job is returned.

Various other systems use similar approaches. For example, the FightAIDS@Home project (http://www.fightaidsathome.org/) utilizes MS Windows PC's in much the same manner as SETI. FightAIDS@Home is even more limited than the SETI project, requiring that the computers have a Windows operating system.

SUMMARY OF THE INVENTION

Embodiments of this invention provide systems and methods for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications. Embodiments utilize proprietary and customized tools to perform efficient computer-aided molecular discovery in a parallel, automated, and platform independent fashion.

One embodiment of this invention includes a method for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the method comprising: (a) receiving an input; (b) providing said input to a first module of a first computer-aided molecular discovery application; (c) providing said input to a second module of a second computer-aided molecular discovery application; (d) executing said first module to create a first output; and (e) executing said second module to create a second output.

Another embodiment of this invention includes a method for integrating a computer-aided molecular discovery process using a plurality of computer-aided molecular discovery applications, the method comprising: (a) receiving an input; (b) providing said input to a first

module of a first computer-aided molecular discovery application; (c) executing said first module to create a first output; (d) providing said first output to a second module of a second computer-aided molecular discovery application; and (e) executing said second module to create a second output.

Another embodiment of this invention includes a system for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the system comprising: an application-neutral computer-aided molecular discovery application framework; a first module interface for a first computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework; and a second module interface for a second computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework.

Another embodiment of this invention includes a computer-readable medium on which is encoded programming code for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the computer-readable medium comprising: (a) program code for receiving an input; (b) program code for providing said input to a first module of a first computer-aided molecular discovery application; (c) program code for providing said input to a second module of a second computer-aided molecular discovery application; (d) program code for executing said first module to create a first output; and (e) program code for executing said second module to create a second output.

Another embodiment of this invention includes a computer-readable medium on which is encoded programming code for integrating a computer-aided molecular discovery process using a plurality of computer-aided molecular discovery applications, the computer-readable medium comprising: (a) receiving an input in a user interface; (b) providing said input to a first module of a first computer-aided molecular discovery application; (c) executing said first module to create a first output; (d) providing said first output to a second module of a second computer-aided molecular discovery application; and (e) executing said second module to create a second output.

Another embodiment of this invention includes a laboratory comprising a system for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the system comprising: an application-neutral computer-aided molecular discovery application framework; a first module interface for a first computer-aided molecular discovery application in communication with said computer-

aided molecular discovery application framework; and a second module interface for a second computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework.

Another embodiment of this invention includes a computer network comprising a system for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the system comprising: an application-neutral computer-aided molecular discovery application framework; a first module interface for a first computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework; and a second module interface for a second computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework.

In one embodiment of this invention, the user is provided with a graphical interface that provides the user with the capabilities of a broad array of application modules, such as calculation engines, from a variety of commercial and custom applications. The calculations may be a model and platform independent. Therefore, implementation of new calculation methods is very simple. One embodiment of this invention is capable of utilizing many different computer platforms, including UNIX and LINUX, and provides load balancing for heterogeneous clusters comprising multiple computing platforms.

Since the system is able to utilize a variety of applications and modules, the system is extremely flexible. The user and/or system administrator chooses the modules to use for performing each task or sub-task.

An embodiment of this invention provides an automated, integrated all-in-one solution. One embodiment includes a queuing system that discriminates job priorities and provides a "divide and conquer" mechanism in job execution. The user interface provides web deployable standardized docking protocols for novice users. One embodiment provides output plate-id information for subsequent screening activities and/or the ability to flexibly include standard drug-like filters. Embodiments of this invention are reconfigurable for any combination of modeling tools.

Also, embodiments of this invention provide enormous benefits in terms of scalability. Each of the processes of the system may be executed in a parallel manner utilizing a heterogeneous cluster of networked computers. These computers may be different in terms of both hardware and operating system from one another. The system determines which nodes of the cluster are available and offloads a portion of the processing for any step to the underutilized node.

The flexibility of embodiments of this invention provides advantages to many different members of the computer-aided molecular discovery market. For example, a laboratory or other organization can increase the efficiency of its scientists, improve the utilization of its computing resources, and easily integrate the variety of applications necessary to perform discovery. Also, by utilizing embodiments of this invention, software developers are able to create custom or commercial modules that can be easily integrated with existing commercial applications. Embodiments of this invention also provide great flexibility to software sellers. The sellers can tout the benefit of multiple commercial applications, which can be integrated under a single easy-to-use interface. System integrators also benefit from utilizing embodiments of this invention. The process of integration becomes much simpler because the integrator is not forced to write various separate applications to integrate each of the various modules a molecular discovery laboratory utilizes.

Further details and advantages of this invention are set forth below.

BRIEF DESCRIPTION OF THE FIGURES

These and other features, aspects, and advantages of the this invention are better understood when the following Detailed Description is read with reference to the accompanying drawings, wherein:

Figure 1 illustrates an exemplary environment for one embodiment of this invention;

Figure 2 illustrates a multi-layer application framework in one embodiment of this invention;

Figure 3 illustrates one embodiment of this invention as a 3-level structure of interrelated modules;

Figure 4 illustrates the general process one embodiment of this invention utilizes in reference to the high-level modules of Figure 3;

Figure 5 illustrates the process implemented by the Protein Sequence Translation module in one embodiment of this invention;

Figure 6 illustrates the binding site hypothesis process in one embodiment of this invention;

Figure 7 illustrates the docking or screening process in one embodiment of this invention;

Figure 8 illustrates the process implemented by the Selection and Analysis module in one embodiment of this invention;

Figure 9 illustrates the general process of presenting and updating the user interface and scheduling and executing jobs in one embodiment of this invention;

Figure 10 illustrates the search process in one embodiment of this invention;

Figure 11 illustrates the general process of creating and executing jobs in one embodiment of this invention;

Figure 12 illustrates utilizing templates and customized jobs in one embodiment of this invention;

Figure 13 illustrates providing email notification of search results in one embodiment of this invention;

Figure 14 illustrates providing modeling results via email in one embodiment of this invention;

Figures 15A & B illustrate providing binding sites results via email in one embodiment of this invention;

Figure 16 illustrates automated docking results via email in one embodiment of this invention;

Figure 17 illustrates the creation and execution of a custom script for a commercial application module in one embodiment of this invention;

Figure 18 illustrates the pre-paralellization process in one embodiment of this invention;

Figure 19 illustrates the paralellization of a process in one embodiment of this invention;

Figure 20 illustrates a scheme for performing *in silico* screening of probes or compounds;

Figures 21A and B are screen shots illustrating an advanced user configuration interface in one embodiment of this invention;

Figure 22 is a screen shot illustrating an administrator configuration interface in one embodiment of this invention;

Figure 23 is a screen shot illustrating a user interface for providing the status of jobs submitted to a heterogeneous cluster in one embodiment of this invention;

Figure 24 is a screen shot illustrating the help system in one embodiment of this invention;

Figure 25 illustrates the process of performing 3-D structure determination in one embodiment of this invention;

Figure 26 illustrates a process for binding site identification in one embodiment of

this invention; and

Figure 27 illustrates a process for docking in one embodiment of this invention.

DETAILED DESCRIPTION OF THE INVENTION

Embodiments of this invention provide systems and methods for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications. One embodiment utilizes one or more modules from various commercial and custom application modules to perform a step in a molecular discovery process, utilizing structure-based, ligand-based, and/or property filter based approaches.

Another embodiment utilizes various application modules to perform multiple steps or tasks in a molecular discovery process, such as the steps that comprise of building homology model(s) and detecting a set of potential binding sites. Yet another embodiment incorporates both horizontal and vertical integration of multiple modules from both commercial and custom applications. Embodiments of this invention may utilize application modules that execute on any hardware / operating system platform and may provide the ability to execute modules in a parallel manner.

In addition, embodiments of this invention may execute any portion of the discovery process in an iterative manner in order to attempt to enhance the results and/or simplify the process for the user. One embodiment of this invention provides a graphical user interface in which a user defines a series of tasks to be performed by one or more application modules. The user also provides the input data necessary for processing. In another embodiment, the user enters information into a file, which is used by the system to perform the discovery process.

The tasks may comprise executing one or many application modules. An input may be provided to a plurality of tasks or the output from one task may provide input to a successive task. The output from a task may be processed by an appropriate script to modify the format as required by the input specification of the successive task. The tasks may perform similar or complementary functions. For example, a task may comprise using one input data set to execute multiple application modules that perform the same general function. The user is then able to compare the resulting output sets.

In one embodiment, a user assembles the tasks within a graphical user interface. Icons or other visual symbols may represent the tasks and the user arranges the tasks in a sequence or flow model to perform the desired discovery process. The interface provides the

user with the capability of entering and/or modifying conditions, thresholds, iteration parameters, and other information necessary to execute the desired process.

In defining the sequence the user may allow interruptions between various tasks so that the user can change the flow as needed. For example, if a user feels that multiple iterations of a task may be needed, the user can set an interruption in the flow after a particular task and decide after each iteration of the task whether to repeat the task or series of tasks until a desired result is achieved or to proceed with subsequent tasks in the process. In addition, the user may desire to repeat or modify particular task(s) using different application module(s). An embodiment of this invention allows such flexibility.

Once a user has optimized the sequence or flow of the process, an embodiment of this invention allows a user to store the sequence or flow for use by other users. Also, the tasks that make up the topology may be executed on a single computer or within a heterogeneous network. Execution in a heterogeneous network may comprise splitting the data and creating scripts to be distributed on the various computing platforms. One embodiment provides the expert users with various prioritization and management facilities.

Referring now to the drawings in which like numerals represent like elements throughout the several figures, Figure 1 illustrates an exemplary environment for one embodiment of this invention utilizing both horizontal and vertical integration as well as parallel execution. In the embodiment shown, a user workstation displays a user interface. The workstation may provide a command line interface, a graphical user interface, or any other interface with which a user may interact. For example, the user interface may comprise a web page, including HyperText Markup Language (HTML), Java, script, and other components. The user interface may also include Tool Command Language (TCL).

A variety of hardware and operating system combinations may support the interface, including Silicon Graphics (SGI) workstations 102, Unix and Linux (*NIX) workstations 104, workstations capable of supporting one of the many flavors of Microsoft Windows 106, and Apple's Macintosh workstations 107.

In the embodiment shown, the user workstation 102-107 accesses an application server 108. The application server 108 may include a web server. In such an embodiment, the web server generates the user interface, accepts parameters from the user interface, and inserts those parameters into a database to, among other purposes, initiate program flow in the application as is discussed in detail below. The application server 108 is able to provide the user interface to a plurality of users concurrently. The application server 108 may also

comprise a plurality of application servers depending on the number of concurrent users anticipated by the systems administrator.

In order to present the user interface and provide various other features, the application server 108 accesses a database. In the embodiment shown, the application server 108 accesses multiple databases, including remote databases 110 and local databases 112, such as control or administrative databases. These databases may include corporate or commercial databases. These databases may be stand-alone databases on a single database server, such as those exemplified by databases 102 and 104, or these databases may include clustered databases 114.

In one embodiment of this invention, the application server 108 uses CGI (Common Gateway Interface), Personalized Homepage Tools (PHP), eXtended Markup Language (XML), and standard data access modules to provide the user interface and process user requests. To initiate jobs, the application server 108 also accesses a computer that executes an application module, such as a server or other member of a heterogeneous cluster 116.

A heterogeneous cluster 116 includes processors. Processors can include, for example, digital logical processors capable of processing input, executing algorithms, and generating output as necessary. Such controllers may include a microprocessor, an Application Specific Integrated Circuit (ASIC), and state machines. Such processors include, or may be in communication with, media, for example computer readable media, which stores instructions that, when executed by the processor, cause the processor to perform the steps described herein as carried out, or assisted, by a processor. Embodiments of computer-readable media include, but are not limited to, an electronic, optical, magnetic, or other storage or transmission device capable of providing a processor, such as the processor in a web server, with computer-readable instructions. Other examples of suitable media include, but are not limited to, a floppy disk, CD-ROM, magnetic disk, memory chip, ROM, RAM, ASIC, configured processor, all optical media, all magnetic tape or other magnetic media, or any other medium from which a computer processor can read. Also, various other forms of computer-readable media may transmit or carry instructions to a computer, including a router, private or public network, or other transmission device or channel.

An application module is a program or portion of a program that can be executed in some manner through the user interface. The module may be an entire commercial application, a single module from a commercial application, a custom module, or some other executable code.

By utilizing variety of application modules to perform calculations, embodiments of this invention operate independently from the constraints of any one commercial application. In addition, it is relatively simple to implement new modules from various applications. Additionally, embodiments of this invention are not limited to operation on a single hardware and software platform. The modules may execute on any platform on which they are designed to function, including *NIX, Microsoft Windows, and other platforms. Not only does this platform independence increase the flexibility of a system according to this invention, it also increases the scalability. One embodiment of this invention is capable of balancing the processing load for performing calculations across heterogeneous clusters, such as heterogeneous cluster 116.

Some commercial applications are only capable of running on a limited number of hardware and operating system environments. For example, one commercially available application may only execute within a Windows environment. An embodiment of this invention does not seek to provide a means for the application to run on hardware or operating systems on which it is not designed to run, but rather to allow the user to control the execution of a module or modules of the application from an integrated user interface.

In the embodiment shown in Figure 1, rather than accessing a single server, the application server 108 accesses a heterogeneous cluster 116 of computers that execute the application module specified by the application server 108. The heterogeneous cluster may include any type and number of computers, both workstations and servers. In the embodiment shown, the heterogeneous cluster includes a rack server 118, the SGI 102 and *NIX 104 workstations, which also may display the user interface, a server cluster 120, and a grid computing architecture 121. An example of the manner in which the application server 108 utilizes the heterogeneous cluster 116 is presented in detail below.

To provide maximum flexibility and scalability, one embodiment of this invention utilizes the multi-layer application framework illustrated in Figure 2 to process requests from the user interface. Figure 2 will now be described with reference to the exemplary environment shown in Figure 1. However, the environment shown in Figure 1 is merely exemplary; the application framework shown in Figure 2 is in no way limited to operating within the environment shown in Figure 1.

The application framework shown in Figure 2 includes a user interface 202 executing on a user workstation, such as an SGI workstation 102. The user interface includes modules 204a-d. The components 204a-d may be presented individually in the user interface 202, such as with component-1 204a and component-2 204b, or be presented in combination

204c,d. When a user specifies a request in the user interface 102, the embodiment shown in Figure 2 executes an "Add Job" process 206. The "Add Job" process 206 creates database records in a table in a database, such as local database 110. For each component 204a-d, multiple "Add Job" processes 206 may execute, creating multiple jobs 208a-d. In addition, in a multi-user environment, each user interface creates independent jobs 208a-d. As jobs 208a-d are created, a "Status" process 209 alerts the user via user workstation 102 or via other means when changes in status of the particular job 208a-d occurs.

In the embodiment shown in Figure 2, a background process or daemon 210 is activated when jobs 208a-d are created in the database 110. The daemon 210 executes the code necessary to create processes within the heterogeneous network 116 corresponding to the job 208a-d. The daemon 210 may be a background process in a *NIX or other environment or may exist as a screen saver in a Microsoft Windows environment or as a process in a different operating environment.

A hypothetical search provides an example of how the process shown in Figure 2 works. A user wishes to search for a protein or nucleic acid structure, so the user enters search criteria in a component 204a-d in the user interface 202. The search request causes the "Add Job" process 206 to add a job 208a-d to database 110. The job 208a-d includes various parameters, including, for example, the sequence, user name, search engines to utilize, and others. The daemon 210 evaluates these parameters and submits the job 208a-d to one or more application modules, search 212 in Figure 2, for processing. The search module 212 performs the necessary processing and then determines whether additional jobs must be performed 218. If so, the "Add Job" process 206 is again executed. If not, a "Notification" process 220 notifies the user that the process is complete. In the example, notification occurs via user workstation 102. However, notification may occur using a variety of methods, including fax, instant messaging, automated phone messaging, e-mail or any other means capable of providing notification to a user. As is shown in Figure 2, one embodiment of this invention may utilize various application modules, including modeling 214 and docking 216 modules. "C- Engines" in Figure 2 refer to calculation engines.

Figure 3 illustrates one embodiment of this invention as a 3-level structure of interrelated modules. The embodiment shown utilizes both horizontal and vertical integration of various application modules as well as the capability of executing various modules in a parallel manner. The embodiment shown integrates visualization, simulation, and application modeling development under the control of a comprehensive user interface 202. The user interface 202 may be a command-line interface, a browser-based interface, or other GUI.

The scientific aspects of the embodiment shown include four broad high-level modules 302-308, which include twelve lower-level modules 312-334. In addition, the embodiment shown also includes an application framework module 310, which includes three lower-level modules 336-340. One embodiment of this invention need not include all of the modules shown in Figure 3. The structure shown is merely illustrative of one embodiment of this invention.

One embodiment of this invention delivers high throughput computer-aided molecular discovery by coupling computational chemistry with high throughput screening. Custom methodology modules can be developed by utilizing tools currently available in the software industry or created independently for data analysis, mining, and visualization. The system may utilize commands, macros, and scripts, allowing applications to be customized by end-users throughout an organization. Also, although embodiments have been described in terms of molecular discovery, the application framework implemented in embodiments of the present invention may be utilized in a variety of applications.

For example, one embodiment of this invention utilizes the following commercially available software packages: Cerius^2 (Accelrys Inc, San Diego, California) and Molecular Operating Environment (MOE™) (Chemical Computing Group, Montreal, Canada) as calculation engines in some of its modules. However, embodiments of this invention are not limited to those or other commercially-available applications. The modular structure of an embodiment allows the implementation of other calculation engines.

The five first-level modules shown in Figure 3 include: (1) a Protein Sequence Translation module 302, which automates the translation of a protein sequence to a three-dimensional structure(s) in an efficient manner (Protein is used only as an example in this specification; any target may be sequenced and ranked in one embodiment of this invention); (2) an Identify Binding Sites module 304, which automates the detection of the putative binding sites, calculates their physico-chemical properties and may perform other functions specified by a user, such as eliminating incorrect sites; (3) a Dock Compounds module 306, which automates the docking of a large number of compounds in an efficient fashion utilizing parallel approaches to split the process among different processors based on protein structures and protein sites and ranks them utilizing a number of scoring functions; (4) a Selection and Analysis module 308, which selects high ranking probes or compounds (Probe and compound are used interchangeably throughout this specification as examples) and submits queries to a database to identify the plates they reside in, analyze them, perform identity, similarity and clustering checks, and rank them for *in biologico* (*in vitro/in vivo*) screening by

generating structure and site specific reports containing plate numbers, location, and the chemical structure of all their constituents; and (5) an Applications Framework module 310, which generates the user interface and provides job control and parallel execution management in the embodiment shown in Figure 3.

As used herein, the term "probe" refers to a molecular framework encompassing association elements suitable for interaction with a macromolecular biological target, such as but not limited to DNA, RNA, peptides, and proteins, said proteins being those such as but not limited to enzymes and receptors.

Figure 4 illustrates a process of *in silico* molecular screening utilized by one embodiment of this invention in reference to the high-level modules of Figure 3. Also illustrated in Figure 4 are exemplary calculation engines that may be applied to each step in the process.

Although the processes shown are described as occurring in a certain order, embodiments of the current invention are not limited to performing a sequential set of modules. The modules may be run in any combination and in any order. Also, each module or group of modules may execute iteratively if a user prefers.

Referring again to Figures 3 and 4, the Protein Sequence Translation module 302 first determines if the submitted sequence corresponds to an existing crystal structure or other experimentally determined three-dimensional structures 402. If not, the three-dimensional structure is determined from the sequence 404. The experimental structure(s) may be retrieved from a protein data bank (www.rcsb.org) or determined using a commercial product, such as but not limited to MOE™ or InsightII® (Accelrys Inc., San Diego, California). Once the three-dimensional structure is determined, or if the crystal structure already exists, the process proceeds to the next step, the binding site hypothesis 406, which is performed by the Identify Binding Sites module 304. A commercial application, such as MOE™, Dock, or Cerius^2, may perform the binding site hypothesis step.

The next step in the process shown is *in silico* screening 408, a step performed by the Dock Compounds module 306. Commercial products, which may be used for this step in the process, include but are not limited to MOE™, Cerius^2, and Schrödinger. This step in the process also retrieves data from a database, such as local database 110. The final step in the *in silico* process is plate selection 410, which is accomplished by the Selection and Analysis module 308. In one embodiment of this invention, plate selection is accomplished via custom code. Once the *in silico* process steps are complete, the compound(s) proceed to *in biologico* screening 412.

Each of the modules of one embodiment of this invention will now be described in detail with reference to Figure 3. The first high-level module is the Protein Sequence Translation module 302. The goal of this module 302 is to automate the creation of a three-dimensional protein model from a protein sequence. Several databases may be used in a concerted fashion to optimize the structural diversity and relevance of the final three-dimensional model that may be used for *in silico* screening, including commercial, public, and proprietary databases. This process is not aimed at substituting for the scientist, but at performing rapid and automated tasks in a way that may not require user's intervention. In one embodiment of this invention, the module 302 generates a series of log files. The scientist has the ability to examine the log files to perform quality control checks and to identify any potential issues. The scientist also has the ability to re-execute (replay) the log file or to modify the log file directly and then execute the series of steps contained in the modified log file as desired.

The embodiment illustrated in Figure 3 is merely exemplary. Other embodiments of this invention include subsets of the modules shown or additional modules. For example, one embodiment of this invention provides links to an integrated data analysis solution. In such an embodiment, information from *in silico* and *in biologico* screening is combined in an integrated user interface. Such an embodiment is described in attorney docket number 41305-283186, filed simultaneously, entitled "System and Method for Data Analysis, Manipulation and Visualization" which is hereby incorporated by reference. Other types of applications may require ad hoc combinations of custom and commercially-available application modules.

Figure 5 illustrates the process implemented by the Protein Sequence Translation module 302. The module 302 first accepts the sequence as an input 502. The module 302 searches for similar sequences commercial and/or proprietary databases and performs multi-sequence alignment 504.

Sequence alignment attempts to align several protein sequences such that regions of structural and/or functional similarity are identified and highlighted. Different matrices are used to perform such alignment, such as but not limited to the freely available engines ClustalW (Jeanmougin, F., Thompson, J. D., Gouy, M., Higgins, D. G. and Gibson, T. J., *Trends Biochem Sci*, **23**, 403-5 (1998)) or MatchBox (Depiereux, E., Baudoux, G., Briffeuil, P., Reginster, I., De Bolle, X., Vinals, C., Feytmans, E., *Comput. Appl. Biosci.* **13(3)** 249-256 (1997)). Databases of protein sequences can be used to identify protein sequences that possess some (user defined) degree of similarity with the protein target of unknown structure,

such as but not limited to the freely available internet-based programs FASTA (http://www.ebi.ac.uk/fasta3/) or BLAST (http://www.ncbi.nlm.nih.gov/BLAST/).

Also, commercially available computer programs, such as but not limited to MOE™ (Chemical Computing Group Inc, Montreal, Canada), Homology (Accelrys Inc., San Diego, California), and Composer™ (Tripos, Inc., St. Louis, Missouri) can perform database searches of the application's proprietary database and sequence alignments as an integrated process. Emphasis can be put on finding similarity among sequences that are known to be associated to certain biological functions, in order to predict not only the structure but also the possible function of the target protein.

The module 302 next selects the highly homologous sequences 506 with known three-dimensional structures and constructs three-dimensional model(s) 508 (homology models). Once construction of the three-dimensional models is complete, the process proceeds to the binding site hypothesis process 406 described in Figure 6.

The process illustrated in Figure 6 begins with the three-dimensional structures output by the Structure Determination from Sequence process 404. These three-dimensional structures are used for binding and/or association site(s) detection 602 (referred to herein as "binding sites"). Once the binding site detection is complete, the binding sites are characterized physically 604. Then the binding sites are ranked 606 and a user-specified number of sites are used for subsequent *in silico* screening. The process then proceeds to *in silico* screening 408.

Referring again to Figure 3, the Protein Sequence Translation module 302 includes three lower-level modules: Retrieve Protein Sequence/Structures 312, Perform Sequence Alignment 314, and Produce 3D Structure 316. In the Retrieve Protein Sequence/Structures module 312, one embodiment of this invention starts from a target sequence and retrieves protein structures that have structural/functional similarity with the target sequence. The module 312 processes the target sequence through a search engine, such as BLAST in NCBI, to search for known protein(s) with similar sequence(s). The module 312 may utilize public sequence and three-dimensional structure databases. In one embodiment, the Retrieve Protein Sequence/Structures 312 performs a search in a database, such as a protein data bank (PDB). In another embodiment of this invention, the user may perform a keyword search. The keywords describe the biological nature of the protein. For example, kinases, and GPCR are keywords that the user may specify. Other modules use the retrieved three-dimensional structures during processing. For example, in the embodiment shown, these three-dimensional protein structures are used to construct a homology model for the target.

Several commercially available computer programs, such as but not limited to MOE™ (Chemical Computing Group Inc, Montreal, Canada), InsightII® (Accelrys Inc., San Diego, California), and Modeler© (Andrej Sali, Rockefeller University, New York, New York, http://guitar.rockefeller.edu/modeller/modeller.html) can be used to perform homology modeling. Threading algorithms are described in Godzik A, Skolnick J, Kolinski A., *J. Mol. Biol.,* 227,227-238 (1992) and in other literature. Commercially available threading software includes MatchMaker™ (Tripos, Inc., St. Louis, Missouri).

The next module in the embodiment shown in Figure 3 is the Perform Sequence Alignment module 314. This module accepts a sequence in a standard format, such as the FASTA format, and searches for proteins of similar sequence in a commercial or other database (e.g. MOE™). The module retrieves these three-dimensional protein structures as well as the three-dimensional protein structures from the previous module 312 and performs a sequence alignment on all of them. The aligned chains, including alignment scores, are passed to the subsequent module.

The Produce 3D Structure module 316 runs a homology model engine for the chain with the highest alignment score, or to that selected by the user and produces a three-dimensional model(s) for the target sequence in Protein Data Bank (PDB) format. In one embodiment, the user may modify the default parameter values of the homology modeling process via user interface 202. The user may also perform quality control checks and may also re-run the same process by selecting an alignment score different that that of the highest score.

In the embodiment shown in Figure 4, the Produce 3D Structure module 316 is the final lower-level module of the Protein Sequence Translation module 302. However, energy minimization and/or molecular dynamics simulations may also be performed for the three dimensional structural model(s) using InsightII (Accelrys Inc., San Diego, California) or MOE™ (Chemical Computing Group, Montreal, Canada) or other application software known to those skilled in the art. The next high-level module is the Identify Binding Sites module 304.

The Identify Binding Sites module 304 includes one lower-level module, the Identify and Rank Binding Sites module 318. This module 318 accepts the three-dimensional model for the target protein and processes it through one of the custom or commercial calculation engines, e.g., Cerius^2. The Identify Binding Sites module 318 uses the calculation engine to identify possible binding sites for the protein and ranks the binding sites by size, saving the first *n* binding sites *(n* specified by the user). These sites are then passed to a specified

calculation engine or engines together with the protein information. The module 318 may utilize additional or other algorithms such as Putative Active Sites with Spheres (PASS: Brady, Jr G.P. and. Stouten P. F.W, *J. Comp. Aided Molec. Design*, 14, 383-401 (2000)) aimed at identifying possible sites as well.

In the case of shape-based methods, the sites are defined based on the shape of the target protein. Within the volume of the target protein, a flood-filling algorithm is employed to search unoccupied, connected grid points, which form the cavities (sites). All sites detected can be browsed according to their size, and a user defined size cutoff eliminates sites smaller than the specified size. Mixed shape/properties sites are defined as connections of hydrophobic and hydrophilic spheres in contact with complementary interacting regions of the target protein. The sites may also be ranked according to the number of hydrophobic contacts made with the receptor, thereby including information about the chemistry of the protein in addition to its geometry.

Once three-dimensional structure(s) of the target protein(s) is (are) obtained, computer programs are used to predict possible drug association sites in these three-dimensional structures. These results are input to the subsequent *in silico* screening process. The Dock Compounds module 306 performs this function and is the next high-level module illustrated in Figure 4. In the embodiment shown, this module 306 uses docking engines in a parallel fashion to screen compound databases or a probe set and so on against protein models to predict compounds that have a higher binding affinity with the protein. Various scoring functions and combinations of scoring functions may then be utilized based on user preferences for scoring the docked protein or compound complex.

Figure 7 illustrates the docking or screening process. The process begins with output from the binding site hypothesis process 406. The parallel optimizer extracts three-dimensional structures of the compounds or probes from a database, such as the local database 110, and prepares the data for parallel processing 702. In the embodiment shown, the data is processed in parallel for both compound structures 704 and identified binding sites 706. Next, automated docking is performed 708. Once the docking is complete, the compounds are ranked according to the scoring function value 710. The docking and ranking information is then output to the plate selection process 410.

As an example of the process shown in Figure 7, in one embodiment, a probe set is treated sequentially and docking can be performed in parallel. For each probe, rotating the bonds of the probe generates a user-defined number of conformers. For example, one thousand (1000) conformers are generated for each probe through a Monte-Carlo procedure.

Other conformational search procedures such as but not limited to simulated annealing, knowledge-based search, systematic conformational search, and others known to one skilled in the art may be employed.

Each of these conformers is docked in an association site using computational methods such as but not limited to those described below. One such method employs the alignment of the non mass-weighted three-dimensional principal moments of inertia of the probes with that of the association site. The conformer is shifted in its best alignment orientation in the association site to improve the docking. The orientation of the conformer that optimizes the fit between the principal moments of inertia of the probe and the association site is saved to disk, the docking score is calculated as described below for that conformer and the docking process repeats with a new conformer of the same probe. Computer programs such as but not limited to "Cerius^2 ® LigandFit" (Accelrys Inc., San Diego), DOCK (University of California at San Francisco), F.R.E.D. (OpenEye Scientific Software, Santa Fe, New Mexico) and others may be used for the docking procedure.

After docking of the conformers, a score is calculated for each of the probe's conformers in the association site. Several scoring functions can be used for that purpose. One such scoring function is described below.

Non-bonded electrostatic interactions and volume exclusion calculations can be performed. In this approach, ΔE, the non-bonded interactions between the probe and the target protein, is calculated from the coulombic and *van der* Waals terms of an empirical potential energy function. ΔE is defined theoretically as: $\Delta E = E(\text{complex}) - [E(\text{Probe}) + E(\text{protein})]$, where E(complex) is the potential energy of the (protein + docked probe) complex, E(probe) is the internal potential energy of the probe in its docked conformation, and E(protein) is the potential energy of the protein alone, *i.e.*, with no probe docked. The protein may be kept fixed during the docking procedure and therefore E(protein) would need to be estimated only once. E(complex) can be calculated either from an explicit description of all the atoms of the protein, or from a grid representation of the association site, the latter being faster in the case where a large number of compounds is to be screened. This approach includes explicitly the calculation of van der Waals interactions between atoms using a Lennard-Jones function. This scoring function favors probes that are small (minimizing van der Waals clashes) and that have large charge-charge interactions between the probe and the protein (maximizing the electrostatic interactions). The scoring function also disfavors probes and/or conformers that exhibit large *van der* Waals clashes between the probes and the protein.

Other scoring functions may be used. These include, but are not limited to LUDI (Böhm, H.J. *J. Comp. Aided Molec. Design*, 8, 243-256 (1994)); PLP (Piecewise Linear Potential, Gehlhaar et al, *Chem. Bio.*, 2, 317-324 (1995); DOCK (Meng, E.C., Shoichet, B.K., and Kuntz, I.D., J. *Comp. Chem.* 13: 505-524 (1992)); and Poisson-Boltzman (Honig, B. et al, *Science*, 268, 1144-9 (1995)).

Some of the above scoring functions are implemented in commercially available software packages such as but not limited to Cerius^2 ® from Accelrys, Inc. (San Diego, California) and MOE™ (Chemical Computing Group Inc., Montreal, Canada)

This docking/scoring process is done independently for each probe. The score calculated for one probe's conformers does not depend on the calculations for other probes. Therefore, this process is highly scalable, and can be distributed among any number of computers that have the required programs. For two computers for instance, the probes can be divided into two groups that will be docked and scored in parallel. Ultimately, each probe could be docked and scored individually on one processor. Massively parallel computer architecture could then be used to linearly improve the efficiency of the process. The docking/scoring approaches described above can be used to perform massive throughput *in silico* screening of compounds.

Referring again to Figure 3, the Dock Compounds module 306 includes various lower level or sub-modules. The first lower-level module is the Calculate Node Load module 320. This module 320 calculates the load for each node on a given heterogeneous cluster. The Divide Data module 322 then divides the data into several pieces to be processed independently on each node in a parallel fashion. For example, in the case of a large structure database (SD) file of chemical structures, the data is divided so that one member of the heterogeneous cluster 116 processes only a portion of the entire data set. Both of these modules 320 & 322 are pre-processing modules; they initiate and launch the tasks necessary to prepare data for docking.

The Create Scripts and Copy Data module 324 is also a pre-processing module. This module 324 (1) executes programs to create per node docking engine scripts and per node shell scripts that ensure data management and proper data allocation and (2) copies the data to the individual nodes. For example, the module 324 creates scripts that are used by later modules to process each portion of the SD file as divided in the preceding module. Once the file is divided into smaller files, each of the smaller files may be copied, such as by FTP (File Transfer Protocol) to the nodes in the heterogeneous cluster 116.

Once pre-processing is complete, the Execute Docking in Parallel module 326 executes. This module 326 executes the docking programs in parallel, i.e., at the same time on different members of the heterogeneous cluster 116. The module 326 may run on any member of the cluster 116, e.g., on the leading node. In particular, the module 326 executes and manages the execution of all the processes created by preceding modules 322-324 until they have all successfully completed.

In the embodiment shown in Figure 3, once pre-processing and docking are complete 320-324, the Perform Post-Processing module 328 executes. This module 328 executes programs for post-processing, including programs that (1) combine the individual SD files after calculation of the *in silico* screening score into one large final SD file, (2) clean up the data on the individual nodes, removing unused files, and (3) perform any additional per node calculation that might be necessary at this point. These modules 322-324 may utilize various formats. For example, to minimize the volume of network traffic utilized by the modules 322-324, the files may be transferred and processed in a compressed format, such as gzip.

The next high-level module in the embodiment shown is the Selection and Analysis module 308. This module includes three lower-level modules: a Select Best Compound(s) module 330, a Retrieve Location Information module 332, and a Perform Similarity Analysis module 334.

Figure 8 illustrates the process implemented by the Selection and Analysis module 308. The process shown in Figure 8 receives output from the *in silico* screening process 408. Based on the ranking process, the best n compounds are selected (wherein n is specified by the user or otherwise) 802. Using identifying information, such as the compound or ID number, plate information is extracted from the database (110) 804. The plates are analyzed by module 806. For example, in one embodiment, additional wells from each plate that are not selected in the *in silico* ranking process, are analyzed to determine if similarities exist with the *in silico* ranked and selected compounds identified in the screening process. These compounds are optionally considered based on their similarity and closeness with the *in silico* ranked compounds. The process iterates for each site 808.

Instead of performing *in biologico* screening on all of the *in silico* probe hits obtained only high-ranking probes may be used for subsequent screening activities. Although it may be more relevant to screen only those probes that are identified as *in silico* probe hits in these plates, various similarity measurements, such as the Tanimoto Coefficient (Tc), may reveal that the other probes in each of the plates containing *in silico* probe hits to be near neighbors. Hence, all the probes contained in all the plates containing *in silico* hit(s) may be subjected to

in biologico screening. Once the plate selection process is complete, the results are used for the *in biologico* screening of the identified and selected compounds or plates 412.

The Selection and Analysis module 308 provides automated selection of chemistry scaffolds. The module 308 also provides automated queries against commercial, public, and proprietary database to select suggested chemistry to be pursued further. In addition, the module 308 provides plate analysis and clustering, providing an indication of confidence in site specificity and identification of scaffolds. The module 308 may also provide automated generation of final reports.

The Select Best Compound(s) module 330 selects the best-ranked conformation for each selected compound. The module 330 next selects the best n compounds or the best m% of all the compounds in their best conformation. The values of n and m may be specified by a system administrator or specified by the user. The module 330 outputs various compound identifiers, such as the compound ID number, so that related information, such as the plate ID number, well ID number, and structure, can be retrieved for each compound.

The Retrieve Location Information module 332 uses the related information to search additional database tables for information, such as the location of the plate identified by the plate ID number. Once a plate has been identified, the information is passed to the next module, the Perform Similarity Analysis module 334. This module 334 may receive information for one or many plates.

The Perform Similarity Analysis module 334 performs similarity analysis between the suggested lists of plates to identify any potentially redundant lists, and provides additional information, such as information to assist in prioritizing list submission for *in biologico* screening. The module 334 also allows for filtering the lists to remove any plate or compound from the list. This feature allows a user to remove a compound from the screening list for any number of reasons, including, for example, the compounds nature or presence in another project. Various other analysis functionalities such as Absorption, Distribution, Metabolism, Excretion, Toxicity (ADMET) or other property filters also be implemented as part of this module.

In the embodiment of this invention illustrated in Figure 3, the modules 302-308 and sub-modules 312-334 described above execute within the application framework described in relation to Figure 2. The application framework is illustrated in Figure 3 as the Application Framework module 310.

The Application Framework module includes three lower-level modules: the Job Scheduling module 336, the User Interface module 338, and the Development Kit module 340.

The Job Scheduling module 336 allows a database such as MySQL or Oracle to be used as a job queuing system for any and all modules of the embodiment shown in Figure 3. The module 336 includes the Add Job 206 and Daemon 210 shown in Figure 2 and may also include wrappers for each module as necessary.

The User Interface module 338 provides the user interface 202. In one embodiment, the module 338 provides a web interface for job submissions, job administration, and viewing of job results. The module 338 may allow cross-platform independence, remote access to job information, and other useful functionality.

The Development Kit module 340 provides the capability to add custom modules to the embodiment illustrated in Figure 3. These modules execute under the application framework as illustrated in Figure 2. They may be written in any of a number of languages, including, for example Perl and C++.

Figure 9 illustrates the general process of presenting and updating the user interface and scheduling and executing jobs in one embodiment of this invention. In the embodiment shown, the interface is a dynamic page named GUI 902. GUI includes top header 904, which includes a dynamic menu module, contentCreator 906. ContentCreator 906 generates web page content based on values passed to the script by a drop-down menu or other user interface element. This script creates all the form elements allowing users to enter information and upload multiple files into the application. The Add2Queue module 910 updates Status 908, which presents status to a user.

The contentCreator 906 accesses the Add2Que module 910 to create jobs. The Add2Que module 910 reads information about the sequence, for example, from a FASTA or other formatted file 912, checks for errors, and utilizes the data along with user parameters supplied from the contentCreator 906 to execute the qAddJob query 914. The qAddJob query 914 inserts records into the local database qDB 110.

qDB 110 in the embodiment shown is a series of database tables that store information on requested job calculations, what type of calculation types are available for a user's site, how to handle each calculation type, and qDaemon 916 parameters for specific computers, including default parameters. qDB 110 is independent of the computer or user requesting a calculation and the computer that will handle the calculation. One function qDB 110 may implement is to store calculation requests, calculation parameters, input and output

data, calculation status, and other information related to requested calculations. Some examples of other information related to a requested calculation include, but is not limited to, who requested the calculation, when the calculation was requested, priority level of the calculation, and searchable user supplied comments related to the requested calculation. The qDB 110 may also store input and output data file information, such as name pattern of the files and how many files, for each calculation type.

qDaemon 916 represents a query executing in a background process waiting for jobs to be inserted into the qDB 110. When a new job is found, qDaemon 916 starts a job 920. Changes to the job table in the database 110 are reflected in GUI 902 via the qStatus 922 and qIDStatus 924 queries.

qDaemon 916 is a precompiled executable daemon that manages calculations running on the computer on which the daemon was started. The qDaemon 916 determines when to start a calculation based on a number of variables including but not limited to time of day and current CPU usage. qDaemon 916 requests information from the qDB 110 for the next calculation job that the daemon can run; the qDB 110 then returns information for the next available valid requested calculation based on a list of valid calculation types given by a qDaemon 916 instance, currently waiting requests, and a priority algorithm. If the calculation type requires input data files from the qDB 110, the qDaemon 916 creates any input data files stored in the qDB 110 in a working directory that is also associated with the calculation that is about to run. The qDaemon 916 then calls a calculation specific wrapper script, based on the calculation type, with the requested calculation parameters. If the calculation type requires data files to be uploaded, the qDaemon 916 uploads the data files to the qDB 110; log files and error log files can be treated as data files.

Valid calculation types that are performed by a particular instance of a qDaemon 916 are determined at initial startup of the daemon via command line or other parameters. Multiple instances of QDaemon 916 may execute on a single computer, supporting multiprocessor computers running multiple non-parallel calculations simultaneously.

Figure 10 illustrates a search process in one embodiment of this invention. The user begins the process shown by starting a search, such as a BLAST search, using a remote, local, commercial party, or custom search utility. The user may also be allowed to select any combination of the available search utilities. The user is also allowed to include results from other searches as an input. In one embodiment of the present invention, Init Search 1002 initiates the BLAST search, PDB file search, or other search programs. If a remote search was chosen, Mirror Search 1006 is executed. If a local search was chosen, Local Search

1010 is executed. If a commercial party search utility was chosen, commercial Party Utility 1004 is executed. If a custom search utility was chosen, then Custom Search Utility 1011 is executed. If the user chooses to use more than one search utility 1002, than the chosen search utilities (1004, 1006, 1010, 1011) will run simultaneously.

Mirror Search 1006 is called for searching remote public database queries. This module mirrors result files to the local server for searching. Local Search is called for searching a local mirrored copy of public databases. This module saves results on the local server. commercial Party Search Utility 1004 is called for commercial party database queries. This module saves results to the local server for searching. Custom Search Utility 1011 is used for queries against a custom database. This saves result files to the local server for searching.

Regardless of which search utilities are chosen, search_all 1012 combines the result files from each search utility. Search_all 1012 then appends the results that the user entered from other searches 1002. Pdb_search 1014 derives unique pdb names from Search_all 1012 and applies other conditions/parameters set by the user 1002 resulting in a list of unique pdb file names. Then download_pdb is called 1016.

Download_pdb 1016 accepts a list of pdb file names and uses the query_PDB module 1018 to query the local pdb database to see if the pdb files exist locally. If the files exist locally the script reports the results to the log file and ends 1020. If the files are not found locally, download_pdb generates requests necessary to download the files using 1022 and then calls updateDB 1024. updateDB 1024 updates the internal database with the names and locations of the downloaded files.

Figure 11 illustrates the general process of creating and executing jobs in one embodiment of this invention. The first step in the process after Start 1101 is the qAddJob process 1102. The qAddJob process 1102 may execute as a result of a command from a user, an automated system event, or any other process or event that results in the creation and execution of a job. The qAddJob process 1102 simply adds records to the qDB database 110. qDaemon 916 is a background process that waits for jobs to be added to the database 110. When jobs are added to the database 110, the qDaemon process 916 evaluates the records and starts the corresponding process.

In the embodiment shown in Figure 11, this process may be one of qSearch 1108, qModel 1110, qSite 1112, qDock 1114, or qSelect 1115. This process is not limited to the five jobs shown. Any other process, such as other 1116, may be executed in this manner with little or no change to the integrated user interface. Thus, one embodiment of this invention

provides great flexibility in the implementation and customization of a computer-aided molecular discovery system.

Figure 12 illustrates utilizing templates and customized jobs in one embodiment of this invention. In the embodiment shown, the first process after Start 1201 is the qAddJob process 1210, which adds a job record to the database, qDB 110. qDaemon 916 again waits for jobs to be added to the database 110. When a job is added, an application template, qTemplate 1202, is executed, which in turn, executes a customized calculation 1204. If additional jobs are spawned from the calculation 1206, another job is simply added to the database, qDB 110, by qAddJob 1210. If not, a notification is sent by some means, such as instant messaging, email, or by another method 1208 .

Figures 13-17 illustrate the process of providing notification, such as by email or other method, of the completion of a job in one embodiment of this invention. As in other aspects of this invention, the qDaemon process 916 waits for jobs to be added to the database, qDB 110. When a job is added, qDaemon 916 begins the appropriate job. In the embodiments shown, the job is one of qSearch 1108, qModel 1110, qSite 1112, qDock 1114, qSelect 1115, or other module process 1116. Each of these jobs executes a corresponding process or series of processes, shown as Init Search through download_PDB 1302, Modelseq 1402, Site 1501, and Dock/Dockrepeat 1504, respectively, in the Figures. Once the process is complete, the notification module 1304 provides notification to a user, such as by email, fax, instant messaging, or other suitable communication method.

Figure 15a illustrates the creation and execution of a custom script for a commercial application module in one embodiment of this invention. In the embodiment shown, the Site process is started 1502 after adding a job to the job database as described above. The execution of the Site process results in the creation of a script, which controls the execution of a third-party commercial, public, or custom application. In Figure 15b, this step is illustrated by the Site.scriptMaker step 1504. This script is then executed in the Site.exe 1506, which executes the calculation engine 1506 necessary to perform calculations for the Site process.

Embodiments of this invention provide many benefits over conventional computer-aided molecular discovery systems and processes. One advantage is the ability to parallelize processes across heterogeneous clusters. Figure 18 illustrates the pre-paralellization process in one embodiment of this invention. The docking process is shown in Figure 18 for purposes of illustration. However, any of the processes of this invention may be parallelized in the same manner. In the embodiment shown, the docking process is started 1802. The

start of the process triggers the parallel process 1804. In order to process the information in parallel, the data file, which is an SD file in the embodiment shown, must be split into multiple smaller files 1806. The process of splitting is performed by an Agent 1808, which is described in detail below. The Agent 1808 next copies the smaller data files to the appropriate node in the heterogeneous cluster 1810. The next process then begins 1812, which is illustrated in Figure 19.

Figure 19 illustrates the paralellization of a process in one embodiment of this invention. The efficient parallelization of the process is achieved through a combination of processes called Agents that pre-process and post-process the tasks required for parallel runs. A global process, Oligarch (1910) manages the actual run of the docking engine on several nodes. The security of the process is insured by appropriate firewall implementations.

Agent is a dynamic process that manages the parallelization of all the tasks involved in *in silico* screening process. Several Agents may handle the pre-processing and the post-processing of the various computational stages in a coherent fashion. As an example, one Agent could be creating input files for the docking engine; another Agent could manage the distribution of all the chemical structures on all the nodes; another Agent could post-process the collection of data.

To perform its function, Agent determines the configuration of the computer cluster by reading a file (input: cluster.conf file) or through some other means. This file contains information about the server name, common directory for that particular machine, calibration data that are used for heterogeneous cluster load balancing. The parallelization process can be used on a heterogeneous Unix/Linux cluster, including SGI machines or SUN or IBM or Linux boxes with different CPU mixes.

Oligarch reads a file describing what programs to run in parallel and runs them simultaneously. Oligarch can be located on any member of the cluster but preferably on the leading node of the cluster. Pre-processing Agents create and distribute programs to be run on each node. When it is done, Oligarch runs and manages the execution of all these processes until they have all successfully completed. After completion, Post-processing Agents post-process the data.

The Dock process as illustrated in Figure 19 provides an illustrative example of the Agents and Oligarch in one embodiment of this invention. The process shown in Figure 19 begins where the process in Figure 18 stops. The data has been divided; in this case a large SD file of chemical structures to be *in silico* screened, into several pieces to be processed

independently on each node in a parallel fashion. Pre-processing Agents 1808a,b initiate and launch tasks and prepare data.

One Agent 1808a creates per node docking engine scripts 1906. Another Agent (not shown) creates per node shell scripts that ensure data management and proper data allocation. One Agent 1808b copies the data to the individual nodes 1908, e.g. in this case the pieces of the original large SD file. Agent 1808b also creates the file that will be used by Oligarch 1910. Oligarch 1910 then executes. After completion, post processing Agent 1808c executes, combining data and copying the data results 1916.

In the embodiment shown, Agent 1808c may actually be multiple Agents. For example, in one embodiment, one Agent combines the individual SD file after calculation of the *in silico* screening score into one large final SD file. One Agent cleans up the data on the individual nodes, removing unused files. One Agent performs any additional per node calculation that might be necessary at this point.

One embodiment of the *in silico* screening method is detailed in the block diagram in Figure 20. Additional detailed aspects of this *in silico* screening method are detailed below. If the molecular target is a protein, the target's sequence (2002) is compared to sequences of proteins of known three-dimensional structures (2003). Multiple sequence alignment (2004) may be performed using sequence threading algorithms, other methods and algorithms known by those skilled in the art, or using methods such as those described below. Sequence alignment attempts to align several protein sequences such that regions of structural and/or functional similarity are identified and highlighted. Different matrices are used to perform such alignment, such as but not limited to the freely available engines ClustalW (Jeanmougin, F., Thompson, J. D., Gouy, M., Higgins, D. G. and Gibson, T. J. *Trends Biochem Sci*, 23, 403-5 (1998)) or MatchBox (Depiereux, E., Baudoux, G., Briffeuil, P., Reginster, I., De Bolle, X., Vinals, C., Feytmans, E *Comput. Appl. Biosci.*13 (3) 249-256 (1997)). Databases of protein sequences can be used to identify protein sequences that possess some (user defined) degree of similarity with the protein target of unknown structure, such as but not limited to the freely available internet-based programs FASTA or BLAST. Commercially available computer programs, such as but not limited to MOE™ (Chemical Computing Group Inc, Montreal, Canada), or Modeler© (Andrej Sali, Rockefeller University, New York, New York, http://guitar.rockefeller.edu/modeller/modeller.html) can perform database searches and sequence alignments as an integrated process. Emphasis can be put on finding similarity among sequences that are known to be associated to certain biological functions, in order to predict not only the structure but also the possible function of the target protein.

Once a protein of known three-dimensional structure (template) has been identified as homologous to the target protein sequence, one or more three-dimensional structures of the target protein may be built (2006) based on the three-dimensional structure of the template using homology modeling techniques known to one skilled in the art.

In homology modeling, one attempts to develop models of an unknown protein from homologous proteins. These proteins will have some measure of sequence similarity and a conservation of folds among the homologues. It is hypothesized that for a set of proteins to be homologous, their three-dimensional structures are conserved to a greater extent than their sequences. This observation has been used to generate models of proteins from homologues with very low sequence similarities.

The steps to creating a homology model may be summarized as follows:

Identifying homologous proteins and determine the extent of their sequence similarity with one another and the unknown;

Aligning the sequences

Identifying structurally conserved and structurally variable regions

Generating coordinates for core (structurally conserved) residues of the unknown structure from those of the known structure(s)

Generating conformations for the loops (structurally variable) in the unknown structure

Building the side-chain conformations

Refining and evaluating the properties of the unknown structure

Several commercially available computer programs, such as but not limited to MOE™ (Chemical Computing Group Inc, Montreal, Canada), InsightII® (Accelrys, Inc., San Diego, California), Homology (Accelrys, San Diego, California), and Composer™ (Tripos, Inc., St. Louis, Missouri) can be used to perform homology modeling. Threading algorithms are described in Godzik A, Skolnick J, Kolinski A. 1992, J Mol Biol 227:227-238 and in other literature. Commercially available threading software includes MatchMaker™ (Tripos, Inc., St. Louis, Missouri).

Several templates can be identified and used to derive one or more three-dimensional structures for the target protein. These different three-dimensional structures for the target protein may be used in a parallel fashion in the *in silico* screening process (2008) described below. Once three-dimensional structure(s) of the target protein(s) is (are) obtained (2006), computer programs are used to predict possible drug binding site(s) (2010) for the compounds in these three-dimensional structures.

Several computer programs can be used to identify possible association site(s) (2010), such as but not limited to the shape-based approach from "Cerius^{2}® LigandFit" (Accelrys Inc, San Diego, California), or the mixed size/properties approach from "MOE™ Site Finder" (Chemical Computing Group Inc., Montreal, Canada).

In the case of shape-based methods, the sites are defined based on the shape of the target protein. Within the volume of the target protein, a flood-filling algorithm is employed to search unoccupied, connected grid points, which form the cavities (sites). All sites detected can be browsed according to their size, and a user defined size cutoff eliminates sites smaller than the specified size. Mixed shape/properties sites are defined as connections of hydrophobic and hydrophilic spheres in contact with mainly hydrophobic regions of the target protein. The sites are ranked according to the number of hydrophobic contacts made with the receptor, therefore including information about the chemistry of the receptor in addition to its geometry.

Possible association sites, once identified using the one or more of the methods described above, are used to perform *in silico* screening (2008) of the probes (2012) or a suitable subset or other compound collections. The screening may be separated into two parts: (i) the docking and (ii) the scoring/ranking (2014) of probes. Both processes may be performed in parallel.

The probe set (2012) is treated sequentially and can be processed in parallel. For each probe, a user-defined number of three-dimensional conformers (2016) are generated by rotating the bonds of the probe. Typically, one thousand conformers are generated for each probe through a Monte-Carlo procedure. Other conformational search procedures such as but not limited to simulated annealing, knowledge-based search, systematic conformational search, and others known to one skilled in the art may be employed.

Each of these conformers is docked in the association site (2008) using computational methods such as, but not limited to, those described below. One such method employs the alignment of the non mass-weighted three-dimensional principal moments of inertia of the probes with that of the association site. The conformer is shifted in its best alignment orientation in the association site to improve the docking. The orientation of the conformer that optimizes the fit between the principal moments of inertia of the probe and the association site is saved to disk, the docking score is calculated (2014) as described below for that conformer and the docking process repeats with a new conformer of the same probe. Computer programs such as but not limited to "Cerius^{2} ® LigandFit" from Accelrys Inc. (San Diego, California), DOCK, (University of California at San Francisco, UCSF), F.R.E.D.

(OpenEye Scientific Software, Santa Fe, New Mexico) and others can be used for the docking procedure.

After docking of the conformers as described above, a score is calculated (2014) for each of the probe's conformers in the association site. Several scoring functions can be used for that purpose. One such scoring function is described below.

In this approach, ΔE, the non-bonded interactions between the probe and the target protein, is calculated from the coulombic and van der Waals terms of an empirical potential energy function. ΔE is defined theoretically as: $\Delta E = E(\text{complex}) - [\ E(\text{Probe}) + E(\text{protein})\ ]$, where E(complex) is the potential energy of the (protein + docked probe) complex, E(probe) is the internal potential energy of the probe in its docked conformation, and E(protein) is the potential energy of the protein alone, *i.e.*, with no probe docked. The protein may be kept fixed during the docking procedure and therefore E(protein) would need to be estimated only once. E(complex) can be calculated either from an explicit description of all the atoms of the protein, or from a grid representation of the association site, the latter being faster in the case where a large number of compounds is to be screened. This approach includes explicitly the calculation of van der Waals interactions between atoms using a Lennard-Jones function. This scoring function favors probes that are small (minimizing van der Waals clashes) and that have large charge-charge interactions between the probe and the receptor (maximizing the electrostatic interactions). The scoring function also disfavors probes and/or conformers that exhibit large van der Waals clashes between the probes and the receptor.

Other scoring functions may be used. These include, but are not limited to LUDI (Böhm, H.J. *J. Comp. Aided Molec. Design*, 8, 243-256 (1994)); PLP (piecewise linear potential, Gehlhaar et al, Chem. Bio., 2, 317-324 (1995); DOCK (Meng, E.C., Shoichet, B.K., and Kuntz, I.D. J. Comp. Chem. 1992 13: 505-524); and Poisson-Boltzman (Honig, B. et al, Science, 268, 1144-9 (1995).

Some of the above scoring functions, are implemented in several commercially available software packages such as but not limited to Cerius^2 ® from Accelrys, Inc. (San Diego, California) and MOE™ (Chemical Computing Group Inc., Montreal, Canada).

Figure 20 illustrates a process for performing a process according to this invention. This docking (2008)/scoring (2014) process is done independently for each probe. The score calculated for one probe's conformers does not depend on the calculations for other probes or conformers. Therefore, this process is highly scalable, and can be distributed among any number of computers that have the required programs. For two computers for instance, the probes can be divided in two groups that will be docked and scored in parallel. Ultimately,

each probe could be docked and scored individually on one processor. Massively parallel computer architecture could then be used to linearly improve the efficiency of the process. The docking (2008)/scoring (2014) approaches described above can be used to perform massive throughput *in silico* screening (2008) of compounds.

Each combination of protein structure and probe conformer may be rank ordered based on the scores calculated as described above. In the present embodiment, the highest-ranking protein structure-probe conformer complexes (based on their scores) are saved for each probe. Optionally, several scoring functions (as described above) may also be utilized yielding a set of scores for each protein structure-probe conformer complex and a consensus score and rank order determined from the set of scores and utilized for the final ranking. Other methods for rank ordering, known to one skilled in the art may also be employed.

The above rank ordered probe list is used to select a subset of probes from the entire probe set to be considered for *in biologico* screening. This subset may be determined using one or more of the following protocols or other protocols known to one skilled in the art.

a) user specified percentage of the rank ordered probe list

b) The first "N" members of the rank ordered probe list, where "N" is the number of probes requested by the user

c) The sample plates containing the probes selected in either protocol a or b

d) The first "M" sample plates containing the probes selected in either protocol a or b where "M" is user specified

e) Optionally, the nearest neighbors of the probes selected in either protocol a or b, where the neighbor selection criteria is user specified (the nearest neighbors of the probes are themselves probes)

f) The sample plates containing the probes selected in protocol e.

g) The first "M" sample plates containing the probes selected in protocol f, where "M" is user specified.

h) A diverse subset of the high ranking probes

i) The corresponding sample plates containing the probe subset from protocol h

In the above protocols, the user specified percentage may typically range from 10 to 60 percent. More preferably between 10 and 50 percent. The number of samples or plates designated as "N" or "M" is dependent on the specific *in biologico* assay, but typically ranges from 1,000 to 100,000 compounds or 10 to 1,000 plates respectively.

The rank ordered probe list (2018) obtained as described above is subjected to *in biologico* screening (not shown) against the target(s). Optionally, the entire probe set (2012),

or a diverse subset (selected using methods known to one skilled in the art) of the entire probe set, or other means of selection (known to one skilled in the art) of a custom subset may be subjected to *in biologico* screening (not shown) against the target(s). The biological activity measured in this screening (described above) is used in the selection of a subset of probes based on a user-selected level of biological activity measured in the *in biologico* screening. This subset of probes is defined as the list of *in biologico* hits (not shown).

Optionally, the nearest neighbors of the *in biologico* hits selected above may be determined using methods for neighbor list selection as described above and subjected to further *in biologico* screening. In the case where one or more near neighbor probe(s) have not been synthesized, they may be synthesized.

The lists of *in silico* and *in biologico* hits are divided into three categories (29410): hits found only *in silico*, hits found only *in biologico*, and hits found both *in silico* and *in biologico*. The members of category are *in silico* hits that are not identified as hits *in biologico*. Conversely, members of category are *in biologico* hits that are not identified as *in silico* hits. The members of category are *in silico* hits that are also identified as *in biologico* hits. A population of category serves to validate the entire process and especially the *in silico* protocols. In practice, a population of 10 percent or more of the selected *in silico* hits (2018) that are also validated through *in biologico* screening is considered to be a strong validation.

The *in biologico* hits populating structural scaffolds/templates may be considered leads and may be optionally utilized in the generation of more complex probes and also included to the Candidate Probe Set.

Optionally, the relative populations of the three categories may be reviewed to determine if there is a need to refine the *in silico* protocols described in Figure 20. In practice, if the *in silico* category contains more than 50 to 60 percent of the *in silico* hits (2018) (the threshold level), refinement is recommended. Likewise, if the second category is populated (the threshold level), refinement is also recommended.

In the case where neighbors of the *in silico* hits and/or the plates containing the *in silico* hits are subjected to *in biologico* screening, the potential arises wherein some of the *in biologico* hits may not have been selected in the *in silico* screening (2018). In this case, the first category may be populated.

Figures 21-27 are screen shots of one embodiment of this invention. The screen shots illustrated in Figures 21-27 are for illustration purposes only. A particular embodiment of this invention may not currently include every feature shown and may include additional

features. For example, the advanced user configuration shown in Figure 21A may not contain all of the applications listed due to licensing constraints, changes in availability, and other factors. Figure 21A is a screen shot illustrating an advanced user configuration interface in one embodiment of this invention. Figure 21B is screen shot illustrating an advanced user configuration interface in an embodiment of this invention used for ligand configuration.

Figure 22 is a screen shot illustrating an administrator configuration interface in one embodiment of this invention. Figure 23 is a screen shot illustrating a user interface for providing the status of jobs submitted to the heterogeneous cluster. Figure 24 is a screen shot illustrating the help system in one embodiment of this invention.

Figure 25 illustrates the process of performing 3-D structure determination. In the embodiment shown, a sequence is entered in the user interface 2502. The sequence is converted to a structure 2504. As described above, multiple sequences may be converted to multiple structures in parallel 2506.

Figure 26 illustrates a process for binding site identification in one embodiment of this invention. In the embodiment shown, the structure 2504 generated according to the process illustrated in Figure 25 is the input. The user utilizes a user interface 2602 to enter parameters to control the identification process. Once the identification is complete, the structure with binding sites identified 2604 is produced. As described above, multiple binding site processes may be performed in parallel 2606.

Figure 27 illustrates a process for docking according to this invention. In the process shown, a three-dimensional model 2702 is input. A user enters docking-related parameters into the user interface 2704. The application modules execute, producing an output, such as the graph 2706 shown in Figure 27. As described above, multiple docking processes may be performed in parallel 2708.

One embodiment of this invention uses a variety of software languages to integrate various modules. For example, in one embodiment of the this invention, Perl is used to perform integration within the user interface; SVL is used for protein modeling; and Cerius^2 and other proprietary and public scripts are used to implement procedures within commercial software packages. Also, shell scripts are implemented where necessary, for example, for parallelization of the process. CGI, PHP, HTML, XML, Java, and JavaScript provide the necessary functionality for presentation with the user interface.

In another embodiment of this invention, ligand based design (LBD) approaches can be used in lieu of or in combination with structure-based design approaches. Furthermore,

several LBD tools may be deployed as illustrated in Fig. 21B. For instance, pharmacophore models could be developed based on the three-dimensional structure(s) of known biologically active compounds using programs such as CATALYST (Accelrys Inc., San Diego, California) or UNITY (Tripos Inc., St. Louis, Missouri). Using these approaches, chemical features aligned in a three-dimensional space are identified and related to biologically activity.

Further refinements can be achieved by combining shape-based filters such as excluded volume and principal moments of inertia with pharmacophore models. Such models can be used to screen TTProbes™ and TTPIntergrated libraries™ (TransTech Pharma, High Point, NC; www.ttpharma.com), commercial or private compound databases, or *in silico* designed compound libraries to identify compounds with similar pharmacophoric features.

In another embodiment, the features of the biologically active reference compound(s) can be represented as fingerprints. These are bit-strings (sequences of 1's (on) or 0's (off)) representing presence or absence of various sub-structural features within a compound's chemical structure. Each bit represents an axis in a multi-dimensional substructure space. Fingerprints may consist of thousands of bits. Thus, a 1000-bit fingerprint represents a point in a 1000- dimensional chemistry space. Similar compounds are expected to be located near each other in this space; dissimilar or "diverse" compounds are expected to be further apart from each other. Thus, biologically active reference compound(s) can be projected in the same chemistry space along with TTProbes™, TTPIntergrated libraries™, commercial or private compound databases, or *in silico* designed compound libraries to identify novel compounds whose biological activities may mimic those of known reference one(s).

Fingerprints are calculated using computer programs available from vendors such as but not limited to MDL Information Systems (San Leandro, California) (ISIS fingerprints) or Daylight Chemical Information Systems Inc. (Mission Viejo, California) (Daylight fingerprints). Similarity metrics such as Tanimoto coefficient (described below) are used to compute inter-compound "distance". The magnitude of this "distance" is directly proportional to the structural similarity between compounds.

Tanimoto coefficient is calculated by $Tc= [Nab] / [Na + Nb - Nab]$, where Na is the number of on-bits in molecule a; Nb the number of on-bits in molecule b, and Nab the number of common on-bits. Identical molecules have Tc of 1. Two compounds are deemed to be similar if they have a Tanimoto coefficient greater than a preset cutoff value. This value depends on the fingerprint used, but is usually 0.8 or above. Computer programs

developed and/or described herein allow the selection of TTProbes™ or TTPIntergrated libraries™, commercial or private databases, or *in silico* designed compound libraries to identify compounds that have a Tc above a user-defined cutoff relative to biologically active reference compound(s).

In another embodiment, common substructures or topology or graph theoretical methods (molecular graphs) are employed to identify TTProbes™ or TTPIntergrated libraries™, commercial or private, or *in silico* designed compound libraries similar to biologically active reference compound(s). Computer programs such as ClassPharmer Suite (Bioreason, Santa Fe, New Mexico) or OEChem (OpenEye Scientific Software, Santa Fe, New Mexico) or implementations of algorithms such as those described by Willet et al (J. Chem. Inf. Comput. Sci., 2002, 42, 305-316) or Schneider et al (Angew. Chem. Int. Ed. Engl., 1999, 38, 2894-2896) or others known to those skilled in the art could also be employed.

In yet another embodiment of the LBD approach, use of the steric and electrostatic fields of biologically active reference compound(s) are used to identify TTProbes™ or TTPIntergrated libraries™, commercial or private, or *in silico* designed compound libraries. Using this technique, quantitative and/or qualitative models that predict biological activity or property of the reference compound(s) can be used to predict biological activities of TTProbes™ or TTPIntergrated libraries™, commercial or private databases, or *in silico* designed compound libraries. Computer software(s) such as COMFA® (Tripos Inc., St. Louis, Missouri) and/or COMSIA developed by Klebe et al (*J. Med. Chem.*, 37, 4130-4146 (1994)) or others known to those skilled in the art could be used for this purpose.

In another embodiment of this invention, Absorption, Distribution, Metabolism, Excretion and Toxicity (ADMET) filters or other chemical functionality/feature filters may be employed to obtain TTProbes™ or TTPIntergrated libraries™, commercial or private databases, or *in silico* designed compound libraries. For instance, a user may be allowed to filter TTProbes™ or TTPIntergrated libraries™, commercial or private databases, or *in silico* designed compound libraries with molecular weight < 500 and/or include compounds that contain at least one basic nitrogen. These chemical functionality/features may be obtained from physicochemical descriptors that could be computed for any given compound using (commercial or private software such as but not limited to MOE™ (Chemical Computing Group, Montreal, Canada) or Cerius^2 (Accelrys Inc., San Diego, California). In addition, flexibility to filter TTProbes™ or TTPIntergrated libraries™, commercial or private databases, or *in silico* designed compound libraries based on ADMET thresholds set by user

could also be achieved. These thresholds could be defined based on experimental measurements available for a series of related compounds or predicted using commercial or private software such as but not limited to iDEA™ (Lion bioscience AG, Heidelberg, Germany) QikProp (Schroedinger Inc., Portland, Oregan) or GastroPlus™ (Simulations Plus, Inc., Lancaster, California). Alternatively, local ADMET models could be developed using any or many of the QSAR/QSPR techniques as implemented in MOE™ (Chemical Computing Group, Montreal, Canada) or Cerius^2 (Accelrys Inc., San Diego, California) or other implementations known to those skilled in the art. These models could then be implemented for predicting the ADMET properties of structurally and chemically related or diverse compounds.

Although structure-based design approaches, ligand-based design approaches, and compound properties filters could be used independently, the embodiments of this invention may combine these approaches in any combinations in performing computer-aided molecular discovery. For instance, the properties filter can be a post-processing filter for the compound subset that comes out of the structure-based design approach results. Alternatively, the properties filter may be employed to screen TTProbes™ or TTPIntergrated libraries™, commercial or private databases, or *in silico* designed compound libraries and the resulting compound subset could then be passed as an input for ligand based design approaches.

Embodiments of this invention may support a variety of functions related to molecular discovery beyond the processes described above. For example, embodiments may support: (1) Large scale (millions) enumeration of library compounds; (2) Parallelized conformation generation; (3) Large scale physico-chemical descriptor and molecular fingerprint calculation; (4) same ligand set, variable protein model analysis; (5) cross-site same protein/variable ligand set analysis; and (6) *in silico* high-throughput screening of compounds, (7) energy minimization and/or molecular dynamics simulations of protein three dimensional structural model(s) and/or protein three dimensional model(s) with ligand bound, (8) ligand based design approaches, (9) property based filters, and (10) QSAR/QSPR modeling.

In addition to the functionality described in detail above, one embodiment of this invention may include a variety of other functions and processes. For example, an embodiment may include administration functions. Various user types are defined, such as administrator, advanced user, and casual or novice user, and the interface and functioning of the system are varied based on the user type.

Some organizations utilizing an embodiment of this invention may require that security measures be implemented to ensure that the data generated and consumed by the system will not become known outside the organization. One embodiment of this invention operates only within a firewall and utilize secured sockets layer to provide security.

One embodiment of this invention may be implemented on a single client site or across multiple client sites, utilizing standard protocols, such as TCP/IP. Therefore, a variety of billing and licensing strategies may be utilized. For example, an organization may purchase an unlimited license, or an organization may simply purchase one or more per-seat licenses. In addition, one embodiment of this invention may be implemented as an application or web service to which organizations subscribe.

The following example provides an illustration of how one embodiment of this invention may be utilized. The process described below was executed by integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications. TTPredict™ provided the automation and integration features necessary to provide both horizontal and vertical integration of the various modules within a heterogeneous computer platform.

Thrombin is a suitable target for drug discovery using this method. Thrombin lies in the final common pathway of coagulation and cleaves fibrinogen to fibrin thereby generating the biological polymer that constitutes part of a blood clot in mammals. Therefore, inhibition of thrombin would be expected to exert an antithrombotic effect.

In the present embodiment, TTProbes™ that are thrombin inhibitors were identified by using TTPredict™ starting from the protein sequence of thrombin. The sequence of thrombin was obtained from Swiss-Prot (www.expasy.ch; Accession number: P00734). This sequence was input to the TTPostGene™ module within TTPredict™.

Using this input protein sequence, TTPostGene™ identified several template protein 3D-structures from the protein data bank belonging to the same family. At this point, the X-ray structure of human thrombin (PDB code: 1AD8) was selected as the template structure to build the homology model using InsightII® (Accelrys Inc., San Diego, California).

Putative binding sites for the homology model built thrombin were identified using Cerius2® (Accelrys Inc, San Diego, California). Of the several binding sites identified by the TTPSite™ module, only the first two sites were utilized for further *in silico* screening efforts.

The two-dimensional structures of the TTProbes™ stored in the database were initially cleaned to remove the salts (if present) and subjected to an energy minimization in order to generate the three-dimensional conformation of the probes. The energy

minimization of the TTProbes™ was performed using MOE™ (Chemical Computing Group, Montreal, Canada) and stored in an SD file. This input compound database containing over 50,000 TTProbes™ was used to perform docking studies against both the identified putative binding sites. For each probe, a maximum of 1000 conformations were generated on the fly using Monte Carlo procedure implemented within Cerius2® (Accelrys Inc, San Diego, California).

The docking of TTProbes™ for each of the binding sites was executed in a parallel process using four Linux nodes and a SGI node. The TTProbes™ were processed in chunks of 1000 probes at a time to a given node and after completion of docking for this chunk, the next 1000 probes were passed so that load balancing between various processors could be achieved.

Each probe conformer was docked into the putative binding site(s) and a score value was assigned for each of the thrombin-related probe conformer complex using the LigScore_Dreiding scoring function. However, only the best scored conformer for each probe was saved. Subsequently, four more scoring functions (PLP1, PLP2, PMF, and JAIN) were employed to score the saved thrombin-related probe conformer complexes for each probe. A correlation matrix obtained for the five scoring functions showed over 80% correlation between PLP1 and PLP2. Consequently, the results of PLP2 were not used or considered further. In addition, several combinations of the scoring functions (usually three) were used to rank order the first 20% of the top ranked probes within each binding site. For instance, if a particular probe was identified in the first 20% of its rank ordered scoring function value and also exhibits similar behavior with respect to two other scoring functions, the probe was assigned a rank of 3. If the probe was identified within the first 20% only in two of the three selected scoring function combinations, then the probe was assigned a rank of 2.

The TTProbe™ ID along with the five scoring function values and also multiple scoring function combination results were passed to TTPSelect™ to select or identify the plate-identifiers. Two thousand of the top ranked unique probes for each scoring functions were identified, labeled as *in silico* probe hits and saved separately, thereby generating 8,000 *in silico* probe hits. Subsequently, the plate identification number containing the *in silico* probe hits along with the number of *in silico* probe hits in each of these plates were obtained.

Instead of performing in biologico screening on the 8,000 *in silico* probe hits obtained by filtering the top two thousand best ranked unique probes using each of the four scoring functions, a subset of the 8,000 *in silico* probe hits were obtained for subsequent screening

activities. A subset of the 800 *in silico* probe hits was achieved by selecting the top ranked plates that contained the maximum number of *in silico* probe hits for each of the scoring functions and submitted for in biologico screening against thrombin.

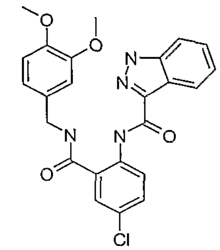
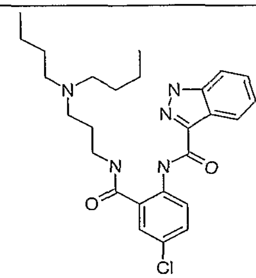
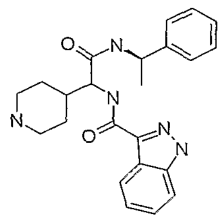
Although it was more relevant to screen only those probes that were identified as *in silico* probe hits in these plates, the computed Tc revealed that the other probes in each of the plates containing *in silico* probe hits to be near neighbors. Hence, all the probes in the selected plates were subjected to in biologico screening against thrombin.

The biological assay for thrombin inhibitory activity is detailed below. To Costar® 96-well black fluorescence plate wells is added 70 microliters of assay buffer, followed by 10 microliters of 1 millimolar substrate solution. Test probe (10 microliters in 30% DMSO) is then added to wells according to the desired concentrations for the assay. The mixture is incubated at 37° C for 5 minutes, followed by addition of 10 microliters of thrombin (100 micrograms/mL in assay buffer), to make a final assay volume of 100 microliters. The plate is mixed gently and incubated 15 minutes at 37° C. Stop buffer (100 microliters) is added, and the plate is read by detecting fluorescence intensity (Excitation at 360 nM, Emission at 460 nM). Percent inhibition of test compound is calculated by comparison with control wells. "Assay buffer" is composed of 100 mM KH2PO4,100 mM Na2HPO4, 1 mM EDTA, 0.01% BRIJ-35, and 1 mM dithiothreitol (added fresh on the day assay is preformed). "Stop buffer" is composed of 100 mM Na-O(O)CCH2Cl and 30 mM sodium acetate which is brought to pH 2.5 with glacial acetic acid. Thrombin was purchased from Sigma (cat #T-3399). Thrombin substrate III fluorogenic was purchased from ICN (cat #195915). Sodium acetate, dithiothreitol, and Brij-35 were purchased from Sigma. Sodium monochloroacetate was purchased from Lancaster 223-498-3. Glacial acetic acid was purchased from Alfa Aesar (cat # 33252). Thrombin was stored at -20°C. Thrombin substrate fluorogenic was stored at - 20° C (5 mM in DMSO).

Based on the dose-response nature of the *in biologico* screened probes, the success of the *in silico* protocols in discovering probes for any given target is exemplified using the *in silico* probe hits that was also identified as an *in biologico* hit, too.

Results are expressed as percentage inhibition at a given test probe concentration in Table 1 below;

TABLE 1

Example	MOLSTRUCTURE	% inhibition @ 100μM	% inhibition @ 50μM
B1		+++	++
B2		+++	++
B3		+++	++

Key

++++	75-100%
+++	40-74%
++	10-39%
+	0-10%

The foregoing description of embodiments of the invention has been presented only for the purpose of illustration and description and is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Numerous modifications and adaptations thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention.

That which is claimed:

1. A method for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the method comprising:

(a) receiving an input;

(b) providing said input to a first module of a first computer-aided molecular discovery application;

(c) providing said input to a second module of a second computer-aided molecular discovery application;

(d) executing said first module to create a first output; and

(e) executing said second module to create a second output.

2. The method of claim 1, wherein receiving said input comprises receiving a sequence.

3. The method of claim 1, wherein said receiving said input comprises receiving a structure.

4. The method of claim 1, wherein receiving said input comprises receiving said input in a user interface.

5. The method of claim 4, wherein receiving said input in said user interface comprises receiving said input in a graphical user interface.

6. The method of claim 4, wherein said receiving said input in a user interface comprises receiving said input in a user interface comprising markup language.

7. The method of claim 4, wherein said receiving said input in a user interface comprises receiving said input in a user interface comprising Tool Command Language (TCL).

8. The method of claim 1, wherein receiving said input comprises receiving said input from a file.

9. The method of claim 1, wherein said first module and said second module are operable to perform a similar function.

10. The method of claim 1, wherein said first module and said second module are operable to perform a complementary function.

11. The method of claim 10, wherein said first module comprises an sequence retrieval module and said second module comprises an alignment module.

12. The method of claim 1, wherein said first computer-aided molecular discovery application comprises a commercially-available application.

13. The method of claim 1, further comprising (f) combining said first output and said second output to create a combined output.

14. The method of claim 13, further comprising (g) presenting said combined output.

15. The method of claim 13, further comprising:

(g) providing said combined output to a third module of a third computer-aided molecular discovery application; and

(h) executing said third module to create a third output.

16. The method of claim15, further comprising:

(i) providing said input to a fourth module of a fourth computer-aided molecular discovery application;

(j) executing said fourth module to create a fourth output; and

(k) combining said third output and said fourth output to create a second combined output.

17. The method of claim 1, further comprising, before step (b), receiving a selection of said first module and a selection of said second module.

18. The method of claim 1, wherein said executing of said first module comprises executing said first module on a heterogeneous computing platform.

19. The method of claim 18, wherein said executing said first module on a heterogeneous computing platform comprises:

determining the load on at least one of a plurality of nodes of said heterogeneous computing platform to identify at least one available node;

creating at least one script for processing on said at least one available node;

performing one or more of the following steps:

dividing a plurality of data elements for processing on said at least one available node,

copying said data to said at least one available node,

copying said at least one script to said at least one available node;

executing said at least one script on said at least one available node; and

combining the output of said execution of said at least one script.

20. The method of claim 18, wherein:

executing said first module comprises executing said first module on a first computing platform; and

executing said second module comprises executing said second module on a second computing platform.

21. The method of claim 1, further comprising, before step (c):

pausing to receive a continuation or cancellation input; and

receiving said continuation or cancellation input.

22. A method for integrating a computer-aided molecular discovery process using a plurality of computer-aided molecular discovery applications, the method comprising:

(a) receiving an input;

(b) providing said input to a first module of a first computer-aided molecular discovery application;

(c) executing said first module to create a first output;

(d) providing said first output to a second module of a second computer-aided molecular discovery application; and

(e) executing said second module to create a second output.

23. The method of claim 22, further comprising:

(f) providing said input to a third module of a third computer-aided molecular discovery application; and

(g) executing said third module to create a third output.

24. The method of claim 23, further comprising:

(h) providing said third output to a fourth module of a fourth computer-aided molecular discovery application; and

(i) executing said fourth module to create a fourth output.

25. The method of claim 24, further comprising:

(j) combining said second output and said fourth output to create a combined output;

(k) providing said combined output to a fifth module of a fifth computer-aided molecular discovery application; and

(l) executing said fifth module to create a fifth output.

26. A system for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the system comprising:

an application-neutral computer-aided molecular discovery application framework;

a first module interface for a first computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework; and

a second module interface for a second computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework.

27. The system of claim 26, wherein said computer-aided molecular discovery application framework comprises a module manager for managing execution of a plurality of modules of a plurality of computer-aided molecular discovery applications within a heterogeneous computing platform.

28. The system of claim 27, wherein said heterogeneous computing platform comprises a grid computing architecture.

29. The system of claim 26, wherein said computer-aided molecular discovery application framework comprises:

a job scheduler;

a paralellization manager; and
a status notifier.

30. The system of claim 29, wherein said paralellization manager comprises:
a pre-processor;
a process manager; and
a post-processor.

31. The system of claim 30, wherein said pre-processor comprises:
a node load manager;
a file splitter; and
a script generator.

32. The system of claim 30, wherein said process manager comprises:
a job database; and
a job daemon.

33. The system of claim 32, wherein said job database comprises a relational database.

34. The system of claim 30, wherein said post-processor comprises:
a file combiner;
a file clean up module; and
a per-node calculation engine.

35. A computer-readable medium on which is encoded programming code for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the computer-readable medium comprising:
(a) program code for receiving an input;
(b) program code for providing said input to a first module of a first computer-aided molecular discovery application;
(c) program code for providing said input to a second module of a second computer-aided molecular discovery application;
(d) program code for executing said first module to create a first output; and
(e) program code for executing said second module to create a second output.

36. The computer-readable medium of claim 35, further comprising:

(f) program code for providing said input to a third module of a third computer-aided molecular discovery application; and

(g) program code for executing said third module to create a third output.

37. The computer-readable medium of claim 36, further comprising:

(h) program code for providing said input to a fourth module of a fourth computer-aided molecular discovery application; and

(i) program code for executing said fourth module to create a fourth output.

38. The computer-readable medium of claim 37, further comprising (f) program code for combining said first output and said second output to create a combined output.

39. The computer-readable medium of claim 38, further comprising (g) program code for presenting said combined output.

40. The computer-readable medium of claim 35, wherein said program code for executing of said first module comprises program code for executing said first module on a heterogeneous computing platform.

41. The computer-readable medium of claim 40, wherein said program code for executing said first module on a heterogeneous computing platform comprises:

program code for determining the load on of at least one of a plurality of nodes of said program code for heterogeneous computing platform to identify at least one available node;

program code for creating at least one script for processing on said at least one available node;

program code for performing one or more of the following steps:

dividing a plurality of data elements for processing on said at least one available node,

copying said data to said at least one available node,

copying said at least one script to said at least one available node;

program code for executing said at least one script on said at least one available node; and

program code for combining the output of said execution of said at least one script.

42. The computer-readable medium of claim 40, wherein:

program code for executing said first module comprises executing said first module on a first computing platform; and

program code for executing said second module comprises executing said second module on a second computing platform.

43. A computer-readable medium on which is encoded programming code for integrating a computer-aided molecular discovery process using a plurality of computer-aided molecular discovery applications, the computer-readable medium comprising:

(a) receiving an input in a user interface;

(b) providing said input to a first module of a first computer-aided molecular discovery application;

(c) executing said first module to create a first output;

(d) providing said first output to a second module of a second computer-aided molecular discovery application; and

(e) executing said second module to create a second output.

44. A laboratory comprising a system for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the system comprising:

an application-neutral computer-aided molecular discovery application framework;

a first module interface for a first computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework; and

a second module interface for a second computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework.

45. A computer network comprising a system for integrating a computer-aided molecular discovery process across a plurality of computer-aided molecular discovery applications, the system comprising:

an application-neutral computer-aided molecular discovery application framework;

a first module interface for a first computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework; and

a second module interface for a second computer-aided molecular discovery application in communication with said computer-aided molecular discovery application framework.

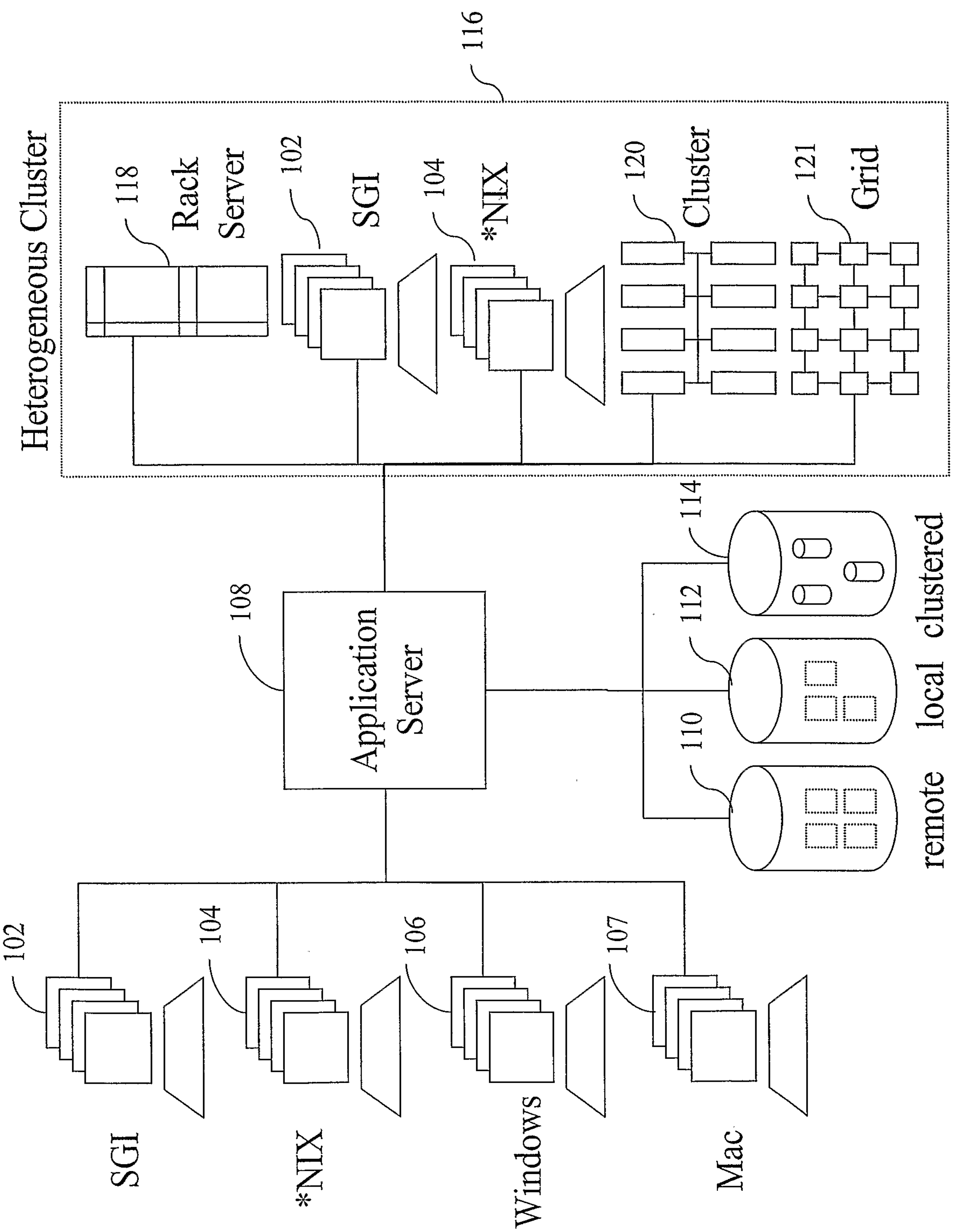

116
Heterogeneous Cluster
118
Rack Server
102
SGI
104
*NIX
120
Cluster
121
Grid
108
Application Server
110
112
114
remote
local
clustered
102
SGI
104
*NIX
106
Windows
107
Mac

FIG. 1

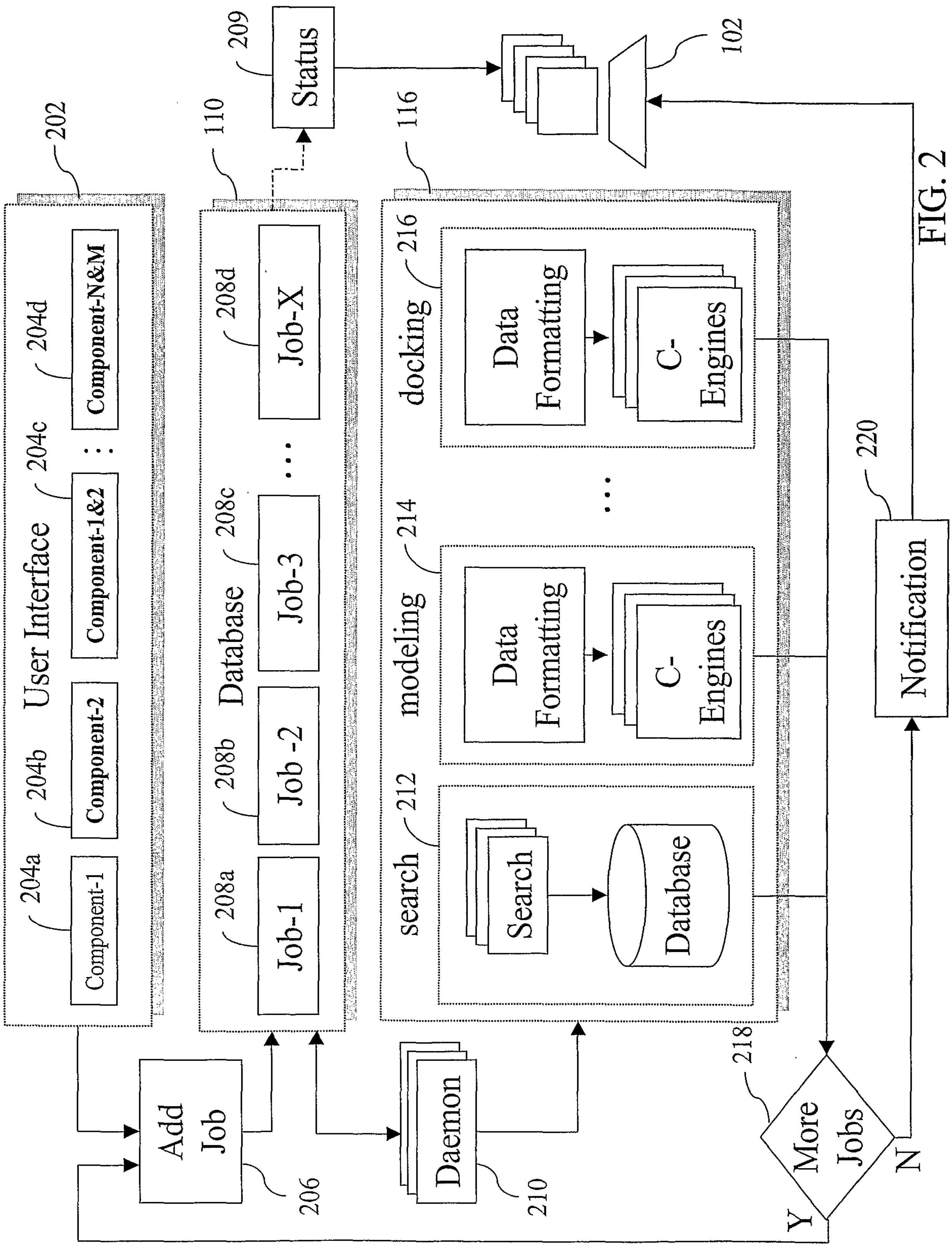

User Interface
Component-1
Component-2
Component-1&2
Component-N&M
202
204a
204b
204c
204d
Database
Job-1
Job -2
Job-3
Job-X
110
208a
208b
208c
208d
Status
209
102
116
search
modeling
docking
212
214
216
Search
Database
Data Formatting
C-Engines
Add Job
206
Daemon
210
More Jobs
218
Y
N
Notification
220

FIG. 2

User Interface 202	Protein Sequence Translation 302	Retrieve Protein Sequence/Structures 312
		Perform Sequence Alignment 314
		Produce 3D Structure 316
	Identify Binding Sites 304	Identify and Rank Binding Sites 318
	Dock Compounds 306	Calculate Node Load 320
		Divide Data 322
		Create Scripts and Copy Data 324
		Execute Docking in Parallel 326
		Perform Post-Processing 328
	Selection and Analysis 308	Select Best Compound(s) 330
		Retrieve Location Information 332
		Perform Similarity Analysis 334
	Application Framework 310	Job Scheduling 336
		User Interface 338
		Development Kit 340

FIG. 3

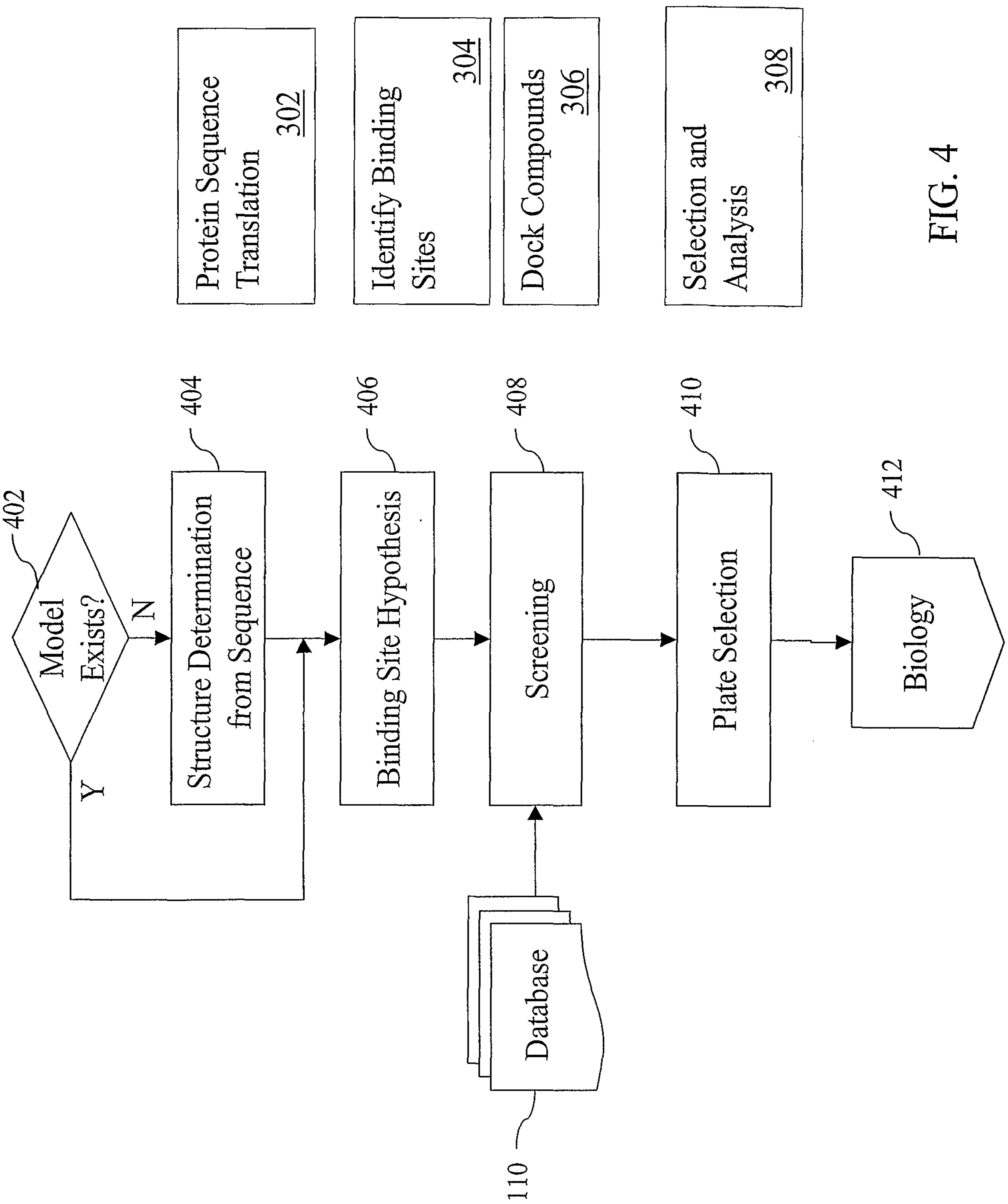

Protein Sequence Translation
302
Identify Binding Sites
304
Dock Compounds
306
Selection and Analysis
308
402
Model Exists?
N
Y
404
Structure Determination from Sequence
406
Binding Site Hypothesis
408
Screening
410
Plate Selection
412
Biology
Database
110

FIG. 4

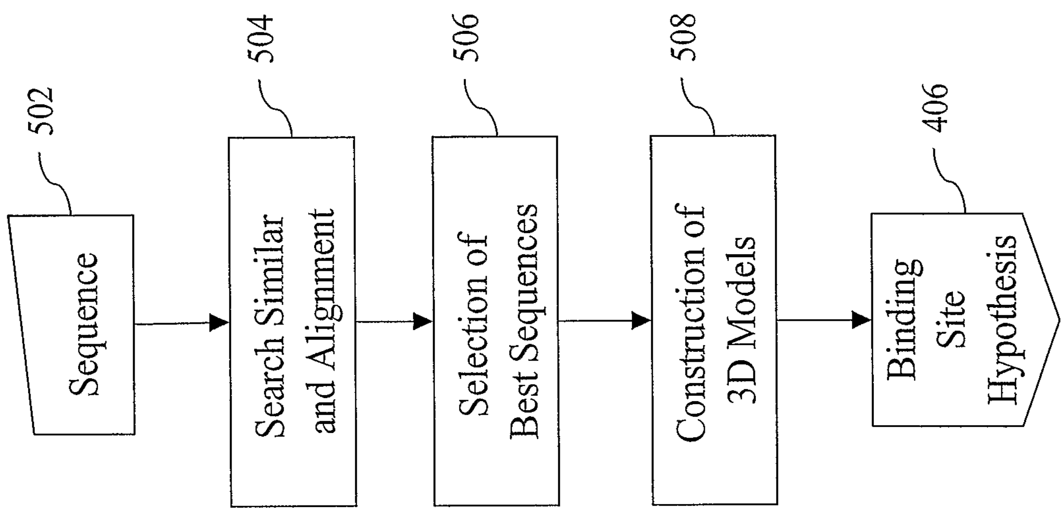

502
Sequence
504
Search Similar and Alignment
506
Selection of Best Sequences
508
Construction of 3D Models
406
Binding Site Hypothesis

FIG. 5

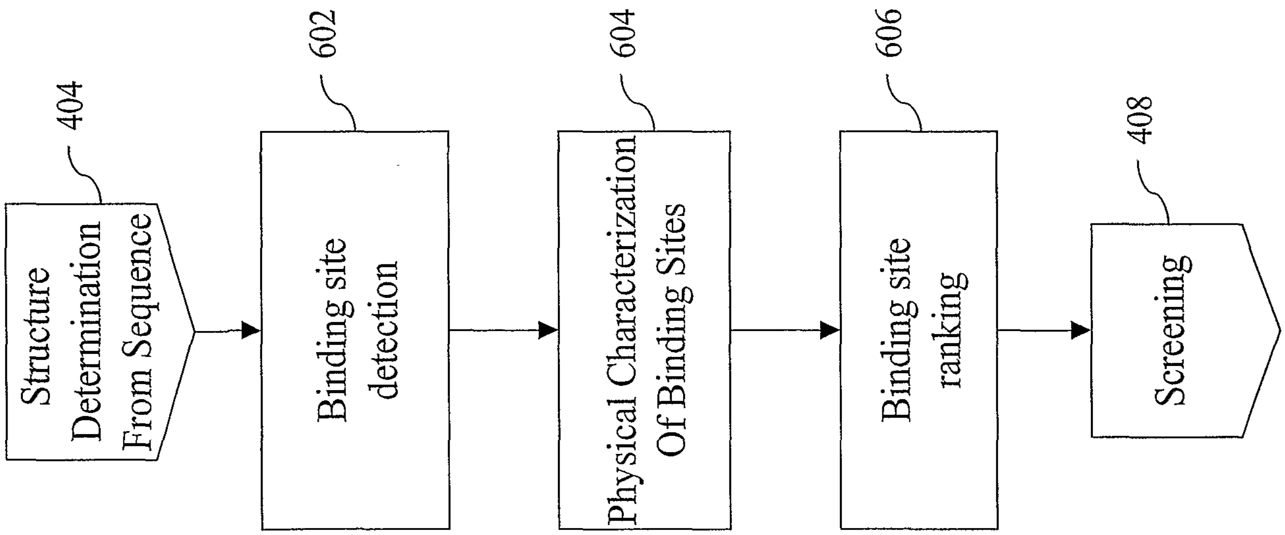

404
Structure Determination From Sequence
602
Binding site detection
604
Physical Characterization Of Binding Sites
606
Binding site ranking
408
Screening

FIG. 6

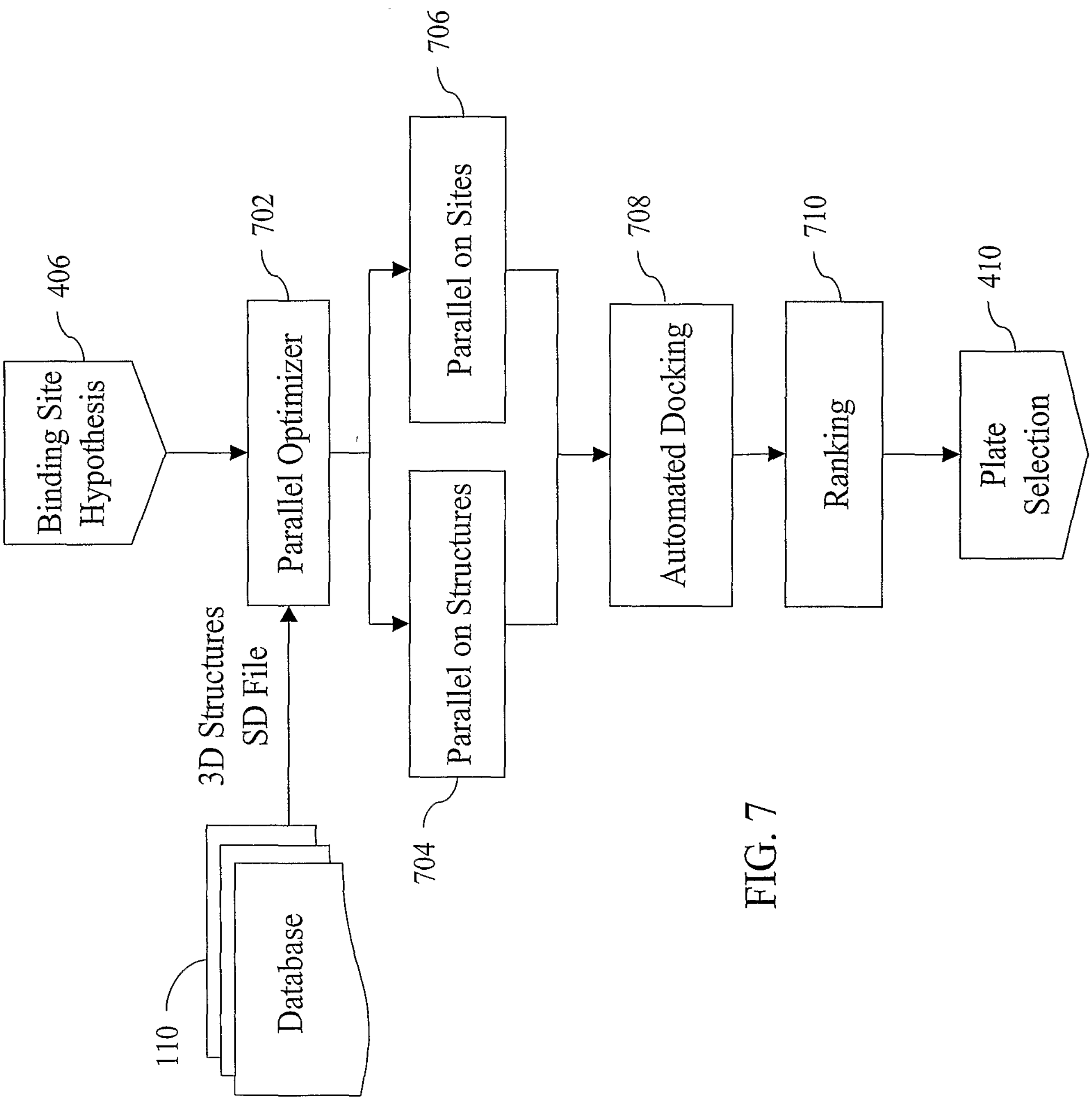

Database
110
3D Structures
SD File
Binding Site
Hypothesis
406
Parallel Optimizer
702
Parallel on Structures
704
Parallel on Sites
706
Automated Docking
708
Ranking
710
Plate
Selection
410

FIG. 7

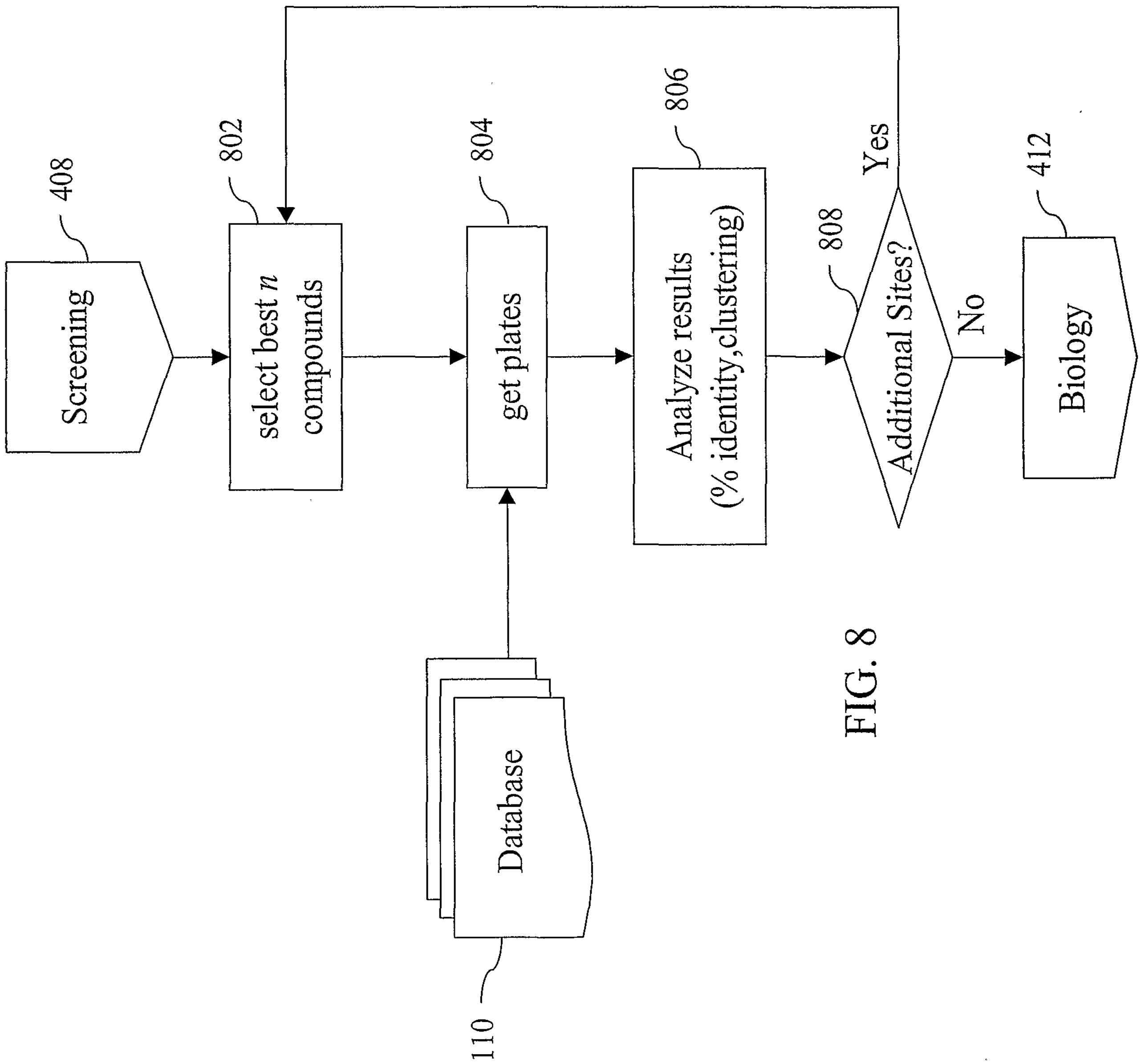

Screening
408
select best n compounds
802
get plates
804
Analyze results (% identity,clustering)
806
Additional Sites?
808
Yes
No
Biology
412
Database
110

FIG. 8

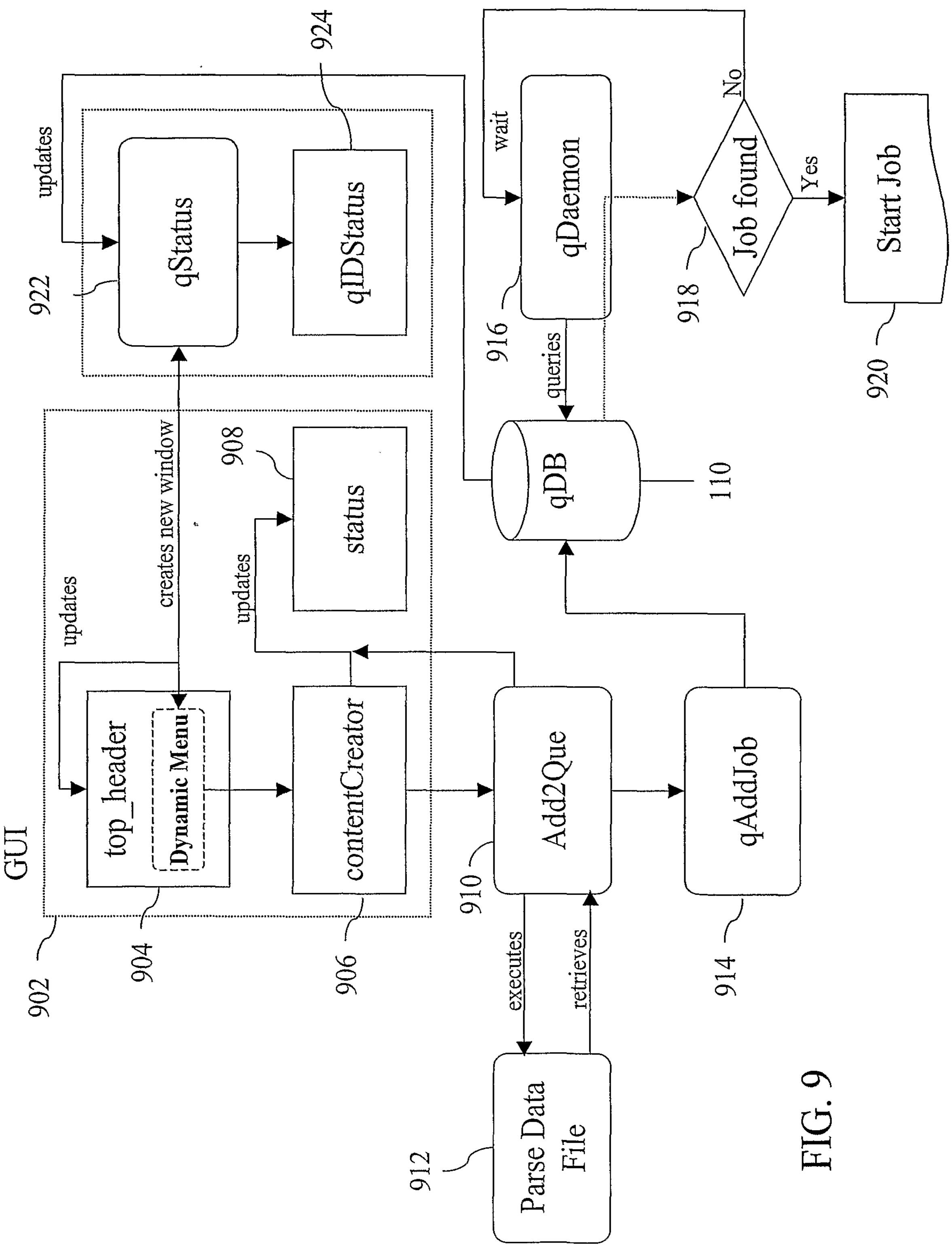

GUI
902
top_header
Dynamic Menu
904
contentCreator
906
status
908
updates
updates
creates new window
qStatus
922
qIDStatus
924
updates
qDB
110
queries
qDaemon
916
wait
Job found
918
No
Yes
Start Job
920
Add2Que
910
executes
retrieves
Parse Data File
912
qAddJob
914

FIG. 9

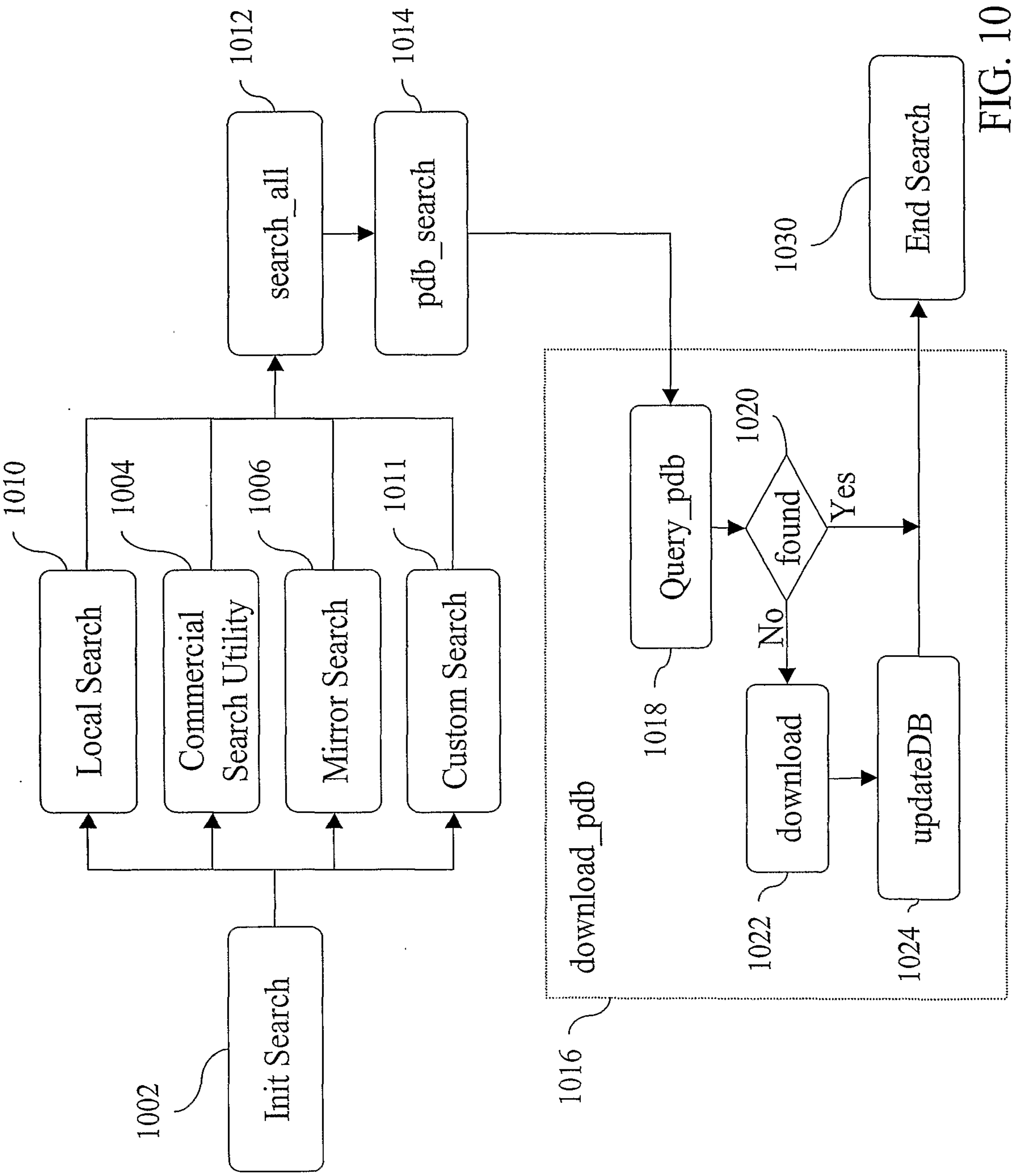

1002
Init Search
1010
Local Search
1004
Commercial Search Utility
1006
Mirror Search
1011
Custom Search
1012
search_all
1014
pdb_search
1016
download_pdb
1018
Query_pdb
1020
found
Yes
No
1022
download
1024
updateDB
1030
End Search

FIG. 10

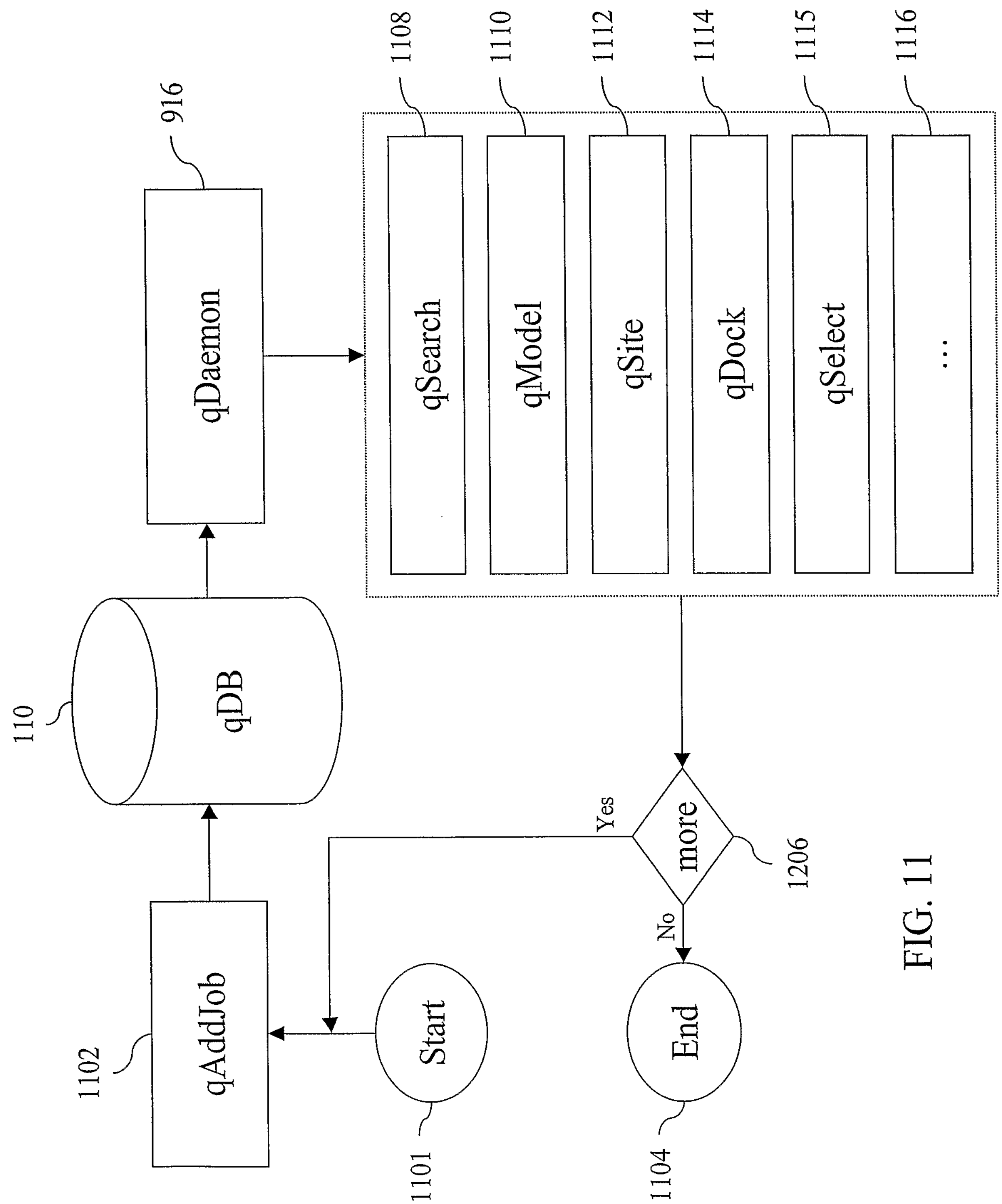

qAddJob
1102
qDB
110
qDaemon
916
qSearch
1108
qModel
1110
qSite
1112
qDock
1114
qSelect
1115
...
1116
Start
1101
more
1206
Yes
No
End
1104

FIG. 11

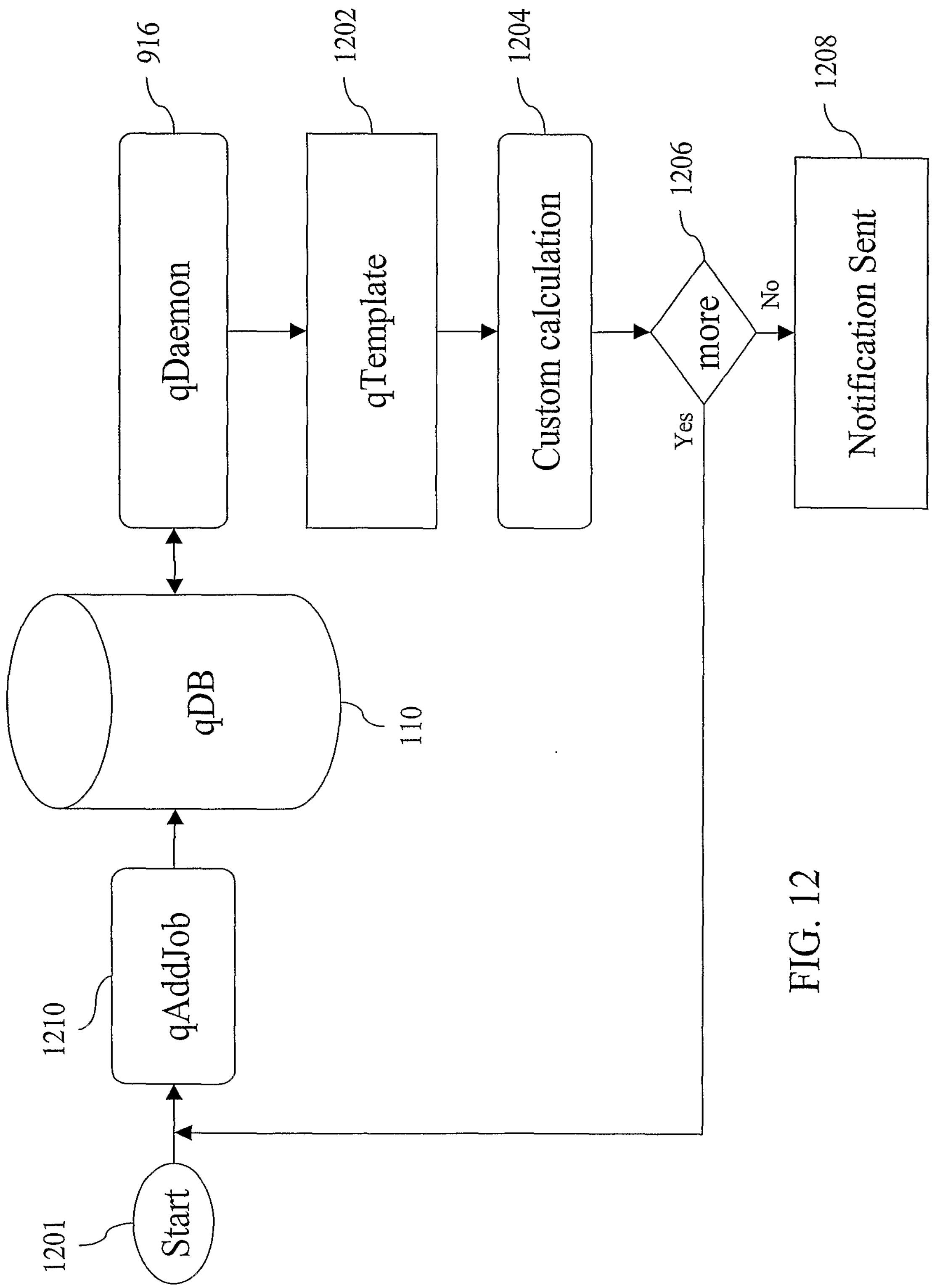

Start
1201
qAddJob
1210
qDB
110
qDaemon
916
qTemplate
1202
Custom calculation
1204
more
1206
Yes
No
Notification Sent
1208

FIG. 12

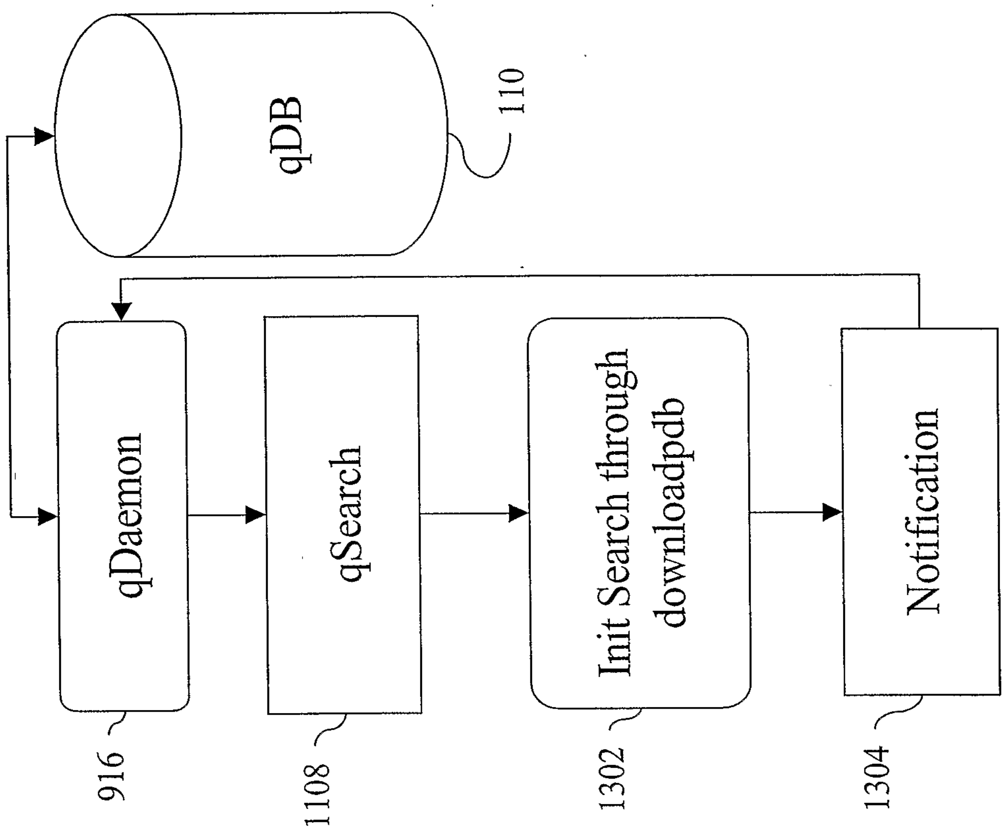

qDB
110
qDaemon
916
qSearch
1108
Init Search through
downloadpdb
1302
Notification
1304

FIG. 13

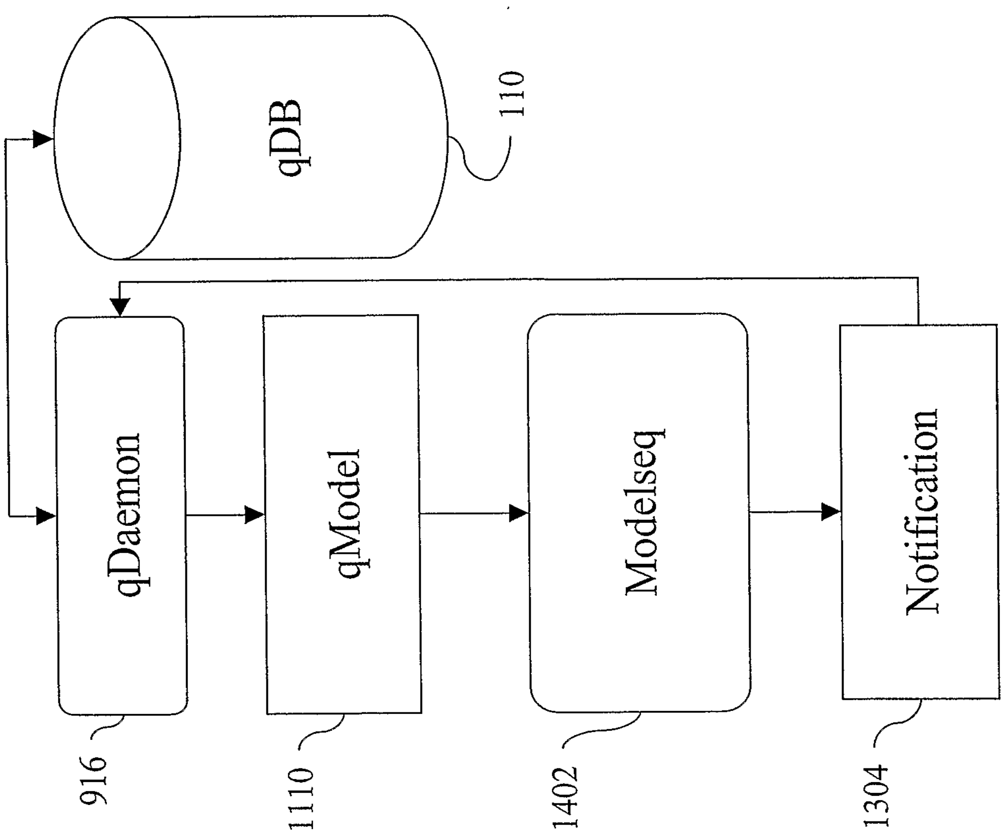

qDB
110
qDaemon
916
qModel
1110
Modelseq
1402
Notification
1304

FIG. 14

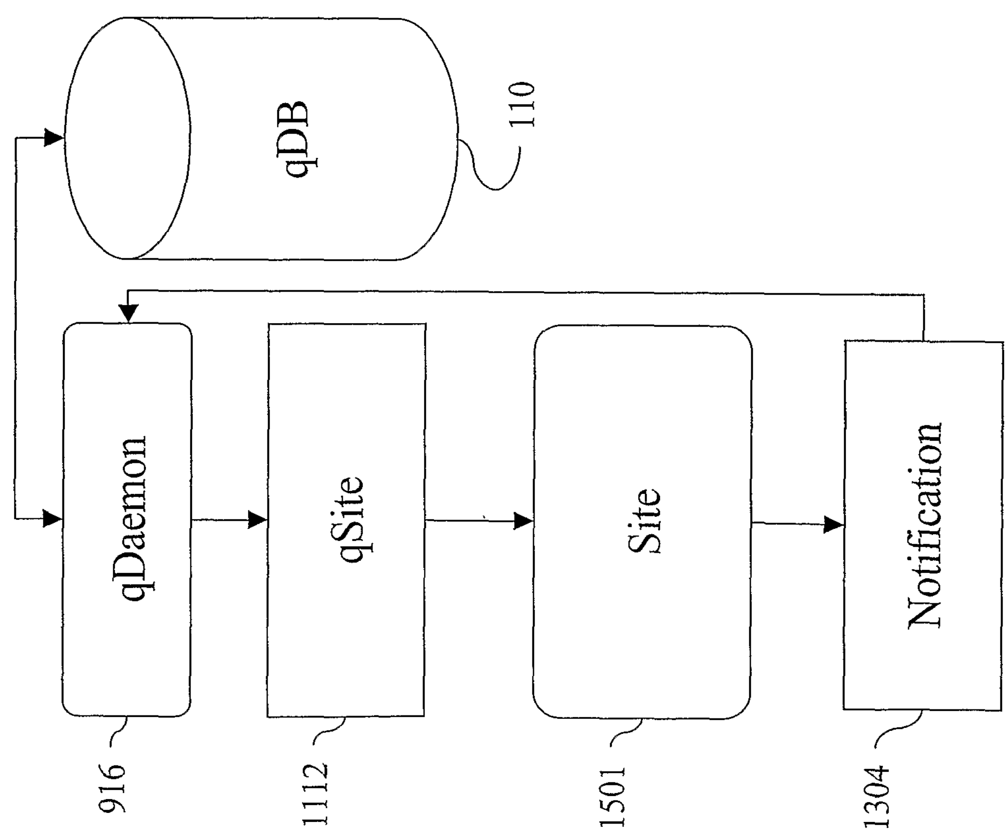


FIG. 15-a

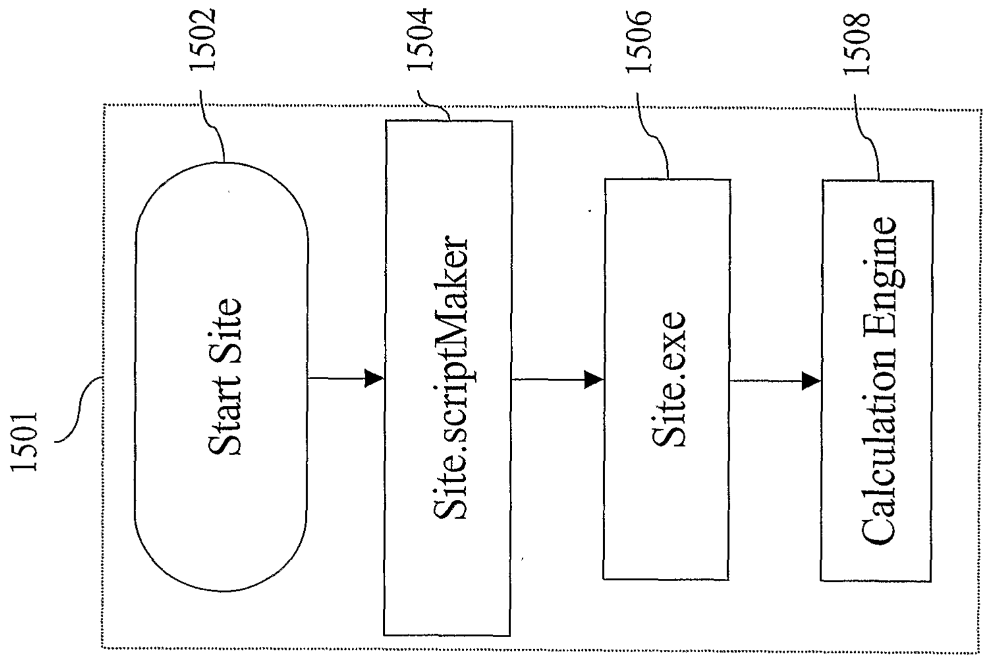


FIG. 15-b

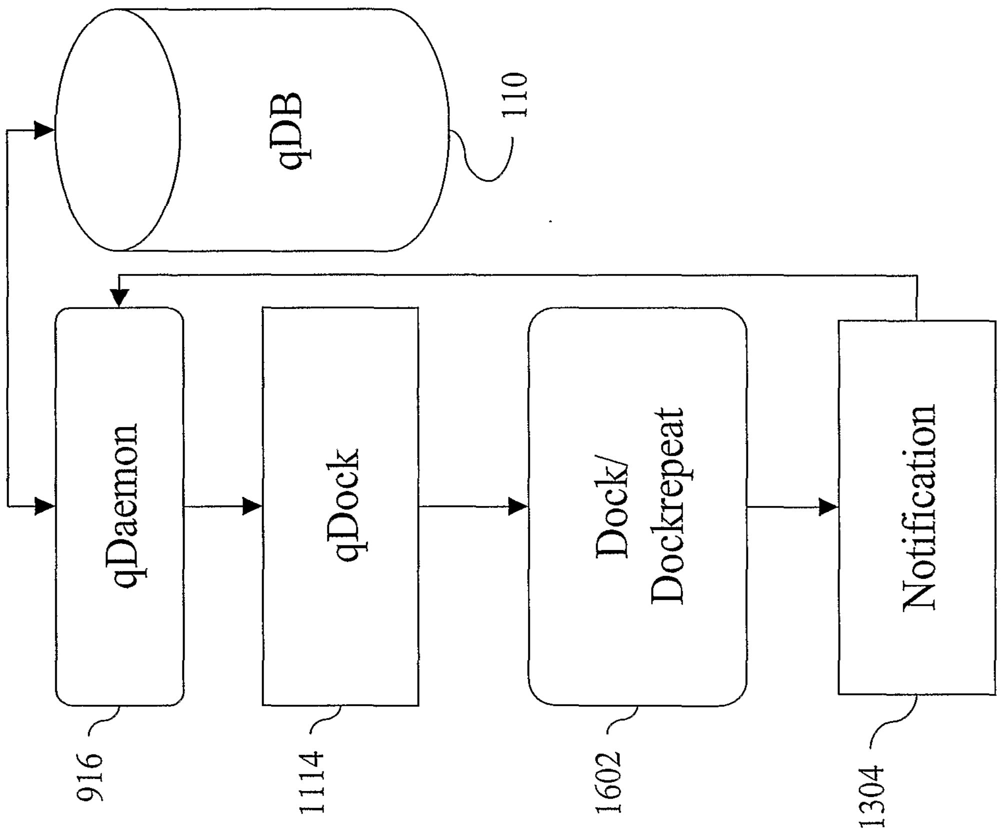

qDB
110
qDaemon
qDock
Dock/
Dockrepeat
Notification
916
1114
1602
1304

FIG. 16

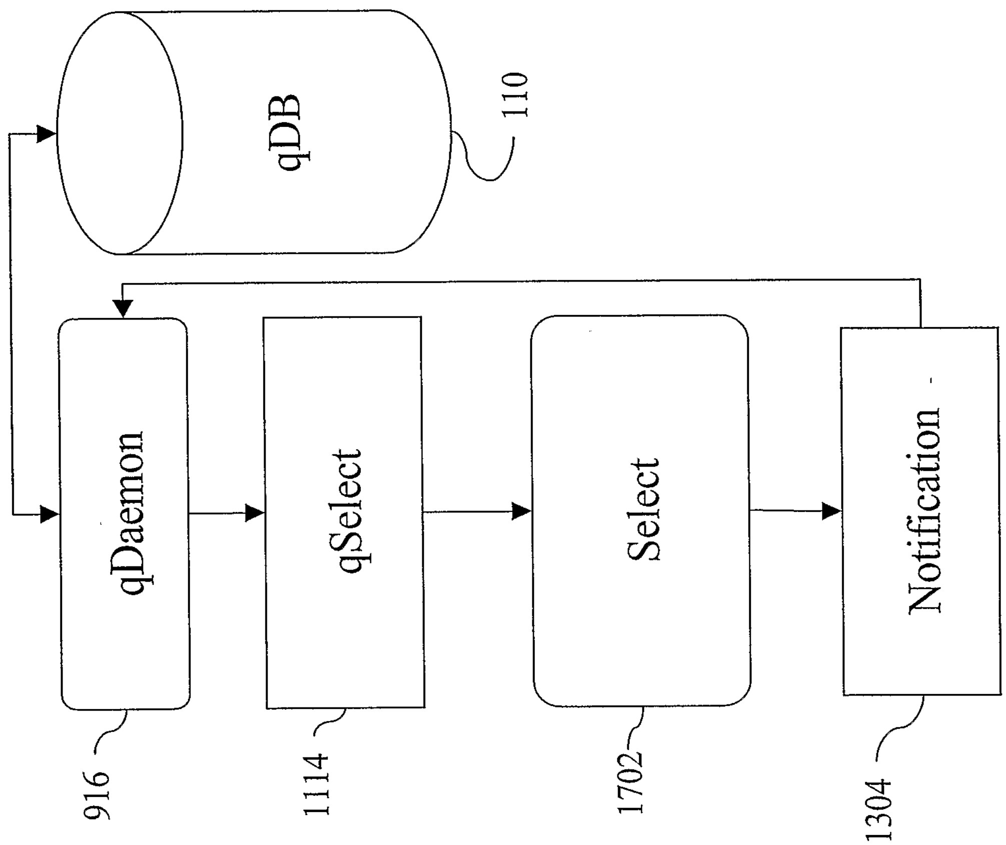

qDB
110
qDaemon
qSelect
Select
Notification
916
1114
1702
1304

FIG. 17

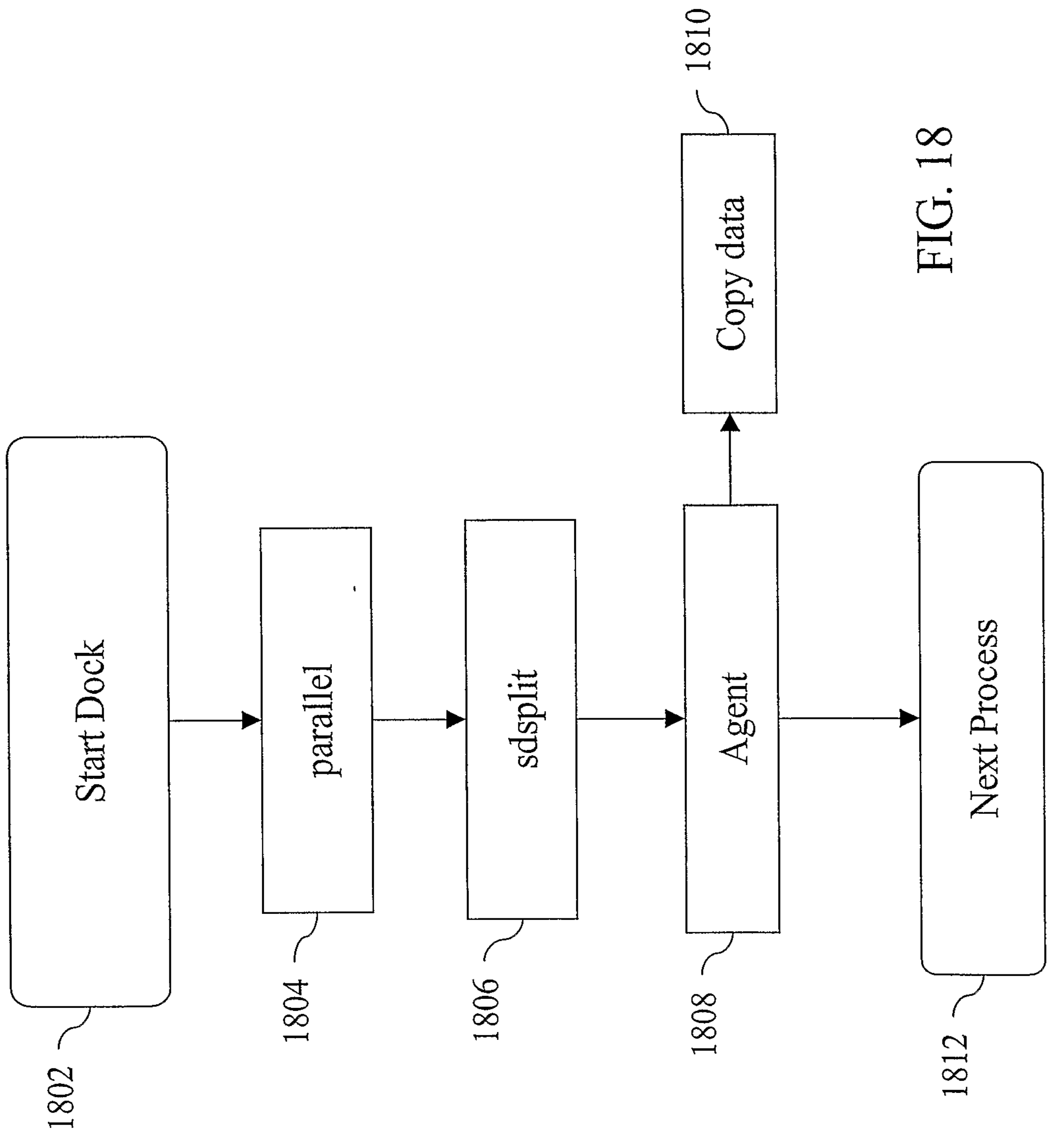

Start Dock
1802
parallel
1804
sdsplit
1806
Agent
1808
Copy data
1810
Next Process
1812

FIG. 18

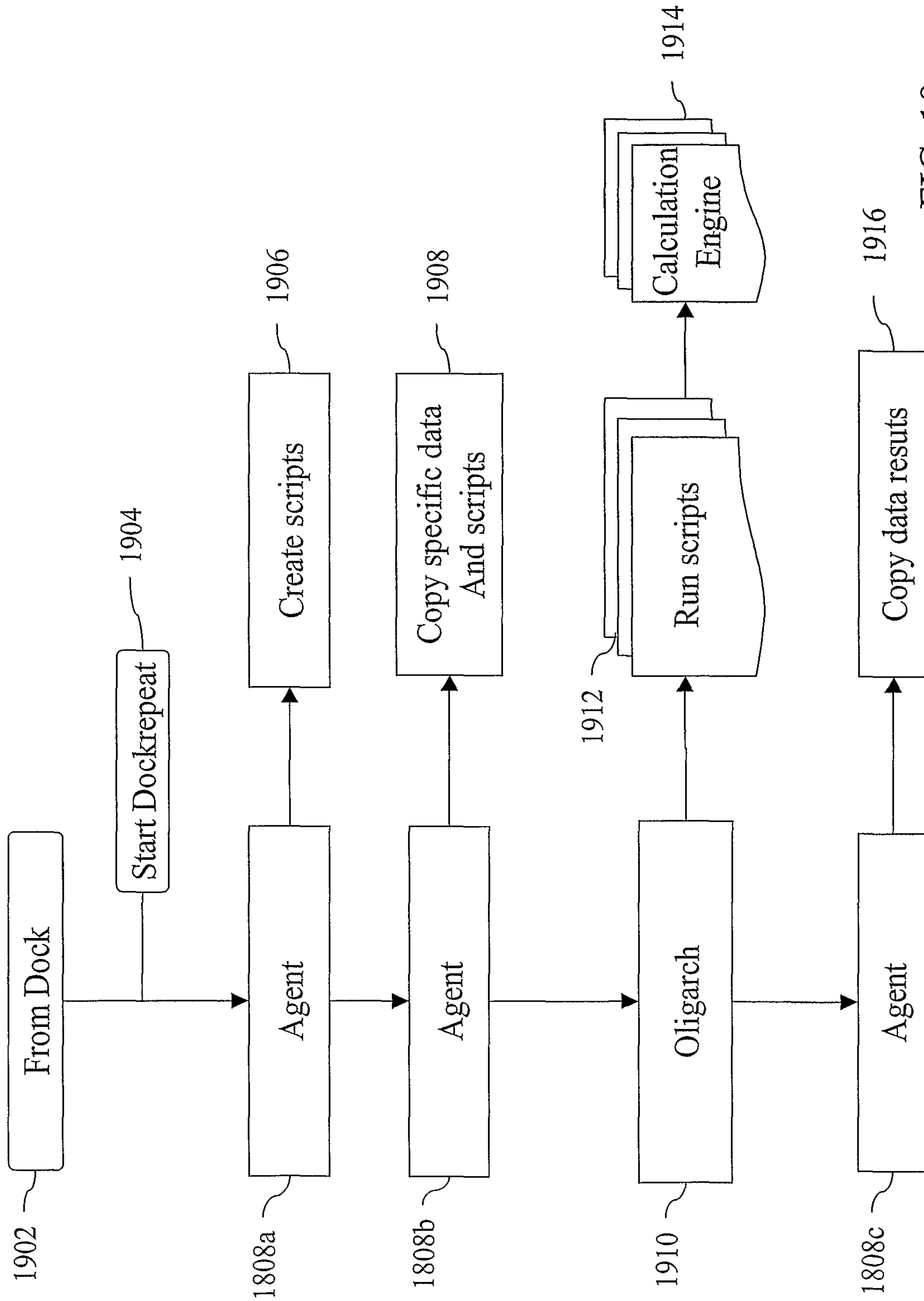

1902
From Dock
1904
Start Dockrepeat
1808a
Agent
1906
Create scripts
1808b
Agent
1908
Copy specific data
And scripts
1910
Oligarch
1912
Run scripts
1914
Calculation
Engine
1808c
Agent
1916
Copy data resuts

FIG. 19

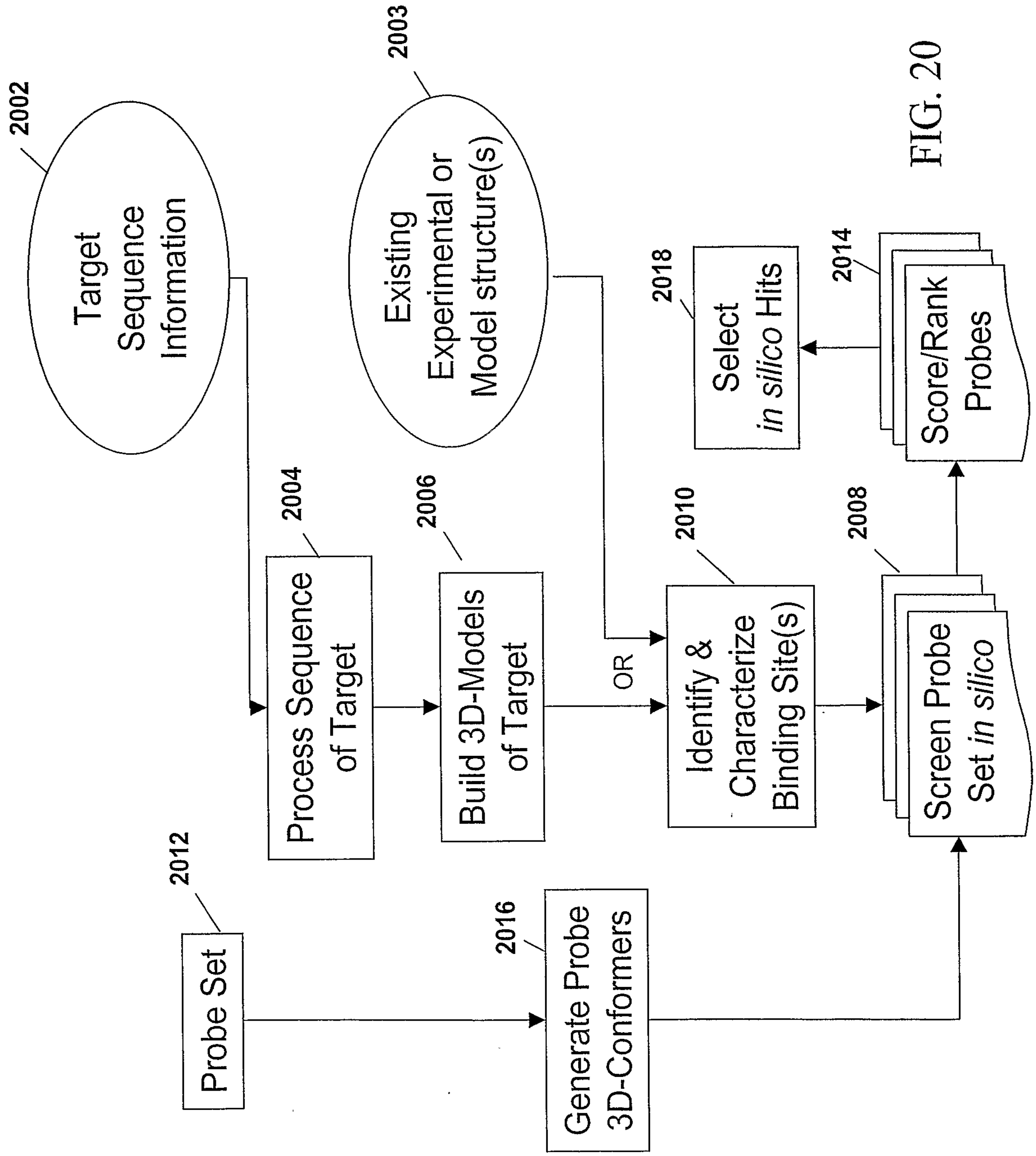

2002
Target Sequence Information
2003
Existing Experimental or Model structure(s)
2004
Process Sequence of Target
2006
Build 3D-Models of Target
OR
2010
Identify & Characterize Binding Site(s)
2012
Probe Set
2016
Generate Probe 3D-Conformers
2008
Screen Probe Set in silico
2014
Score/Rank Probes
2018
Select in silico Hits

FIG. 20

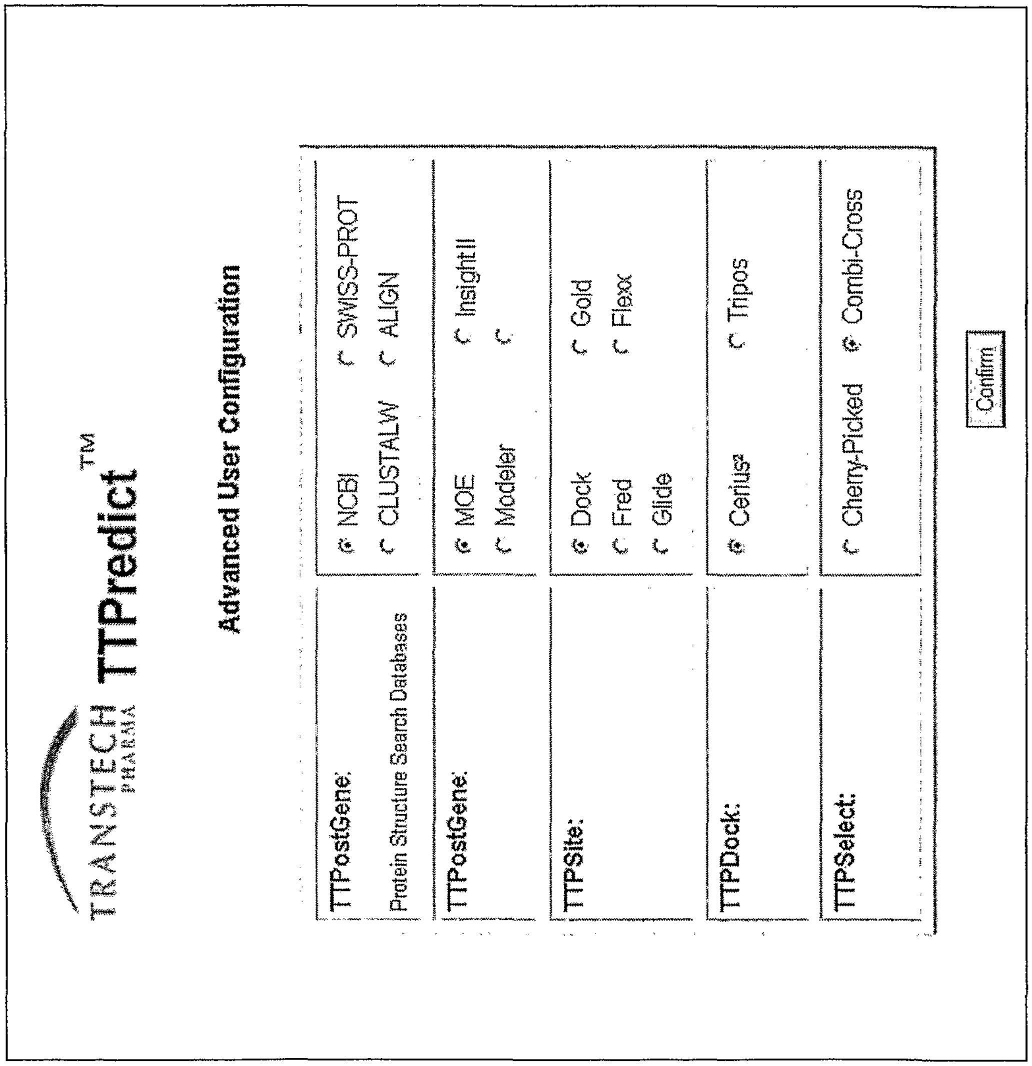

TRANSTECH
PHARMA
TTPredict™
Advanced User Configuration
TTPostGene:
Protein Structure Search Databases
NCBI
SWISS-PROT
CLUSTALW
ALIGN
TTPostGene:
MOE
InsightII
Modeler
TTPSite:
Dock
Gold
Fred
Flexx
Glide
TTPDock:
Cerius²
Tripos
TTPSelect:
Cherry-Picked
Combi-Cross
Confirm

FIG. 21a

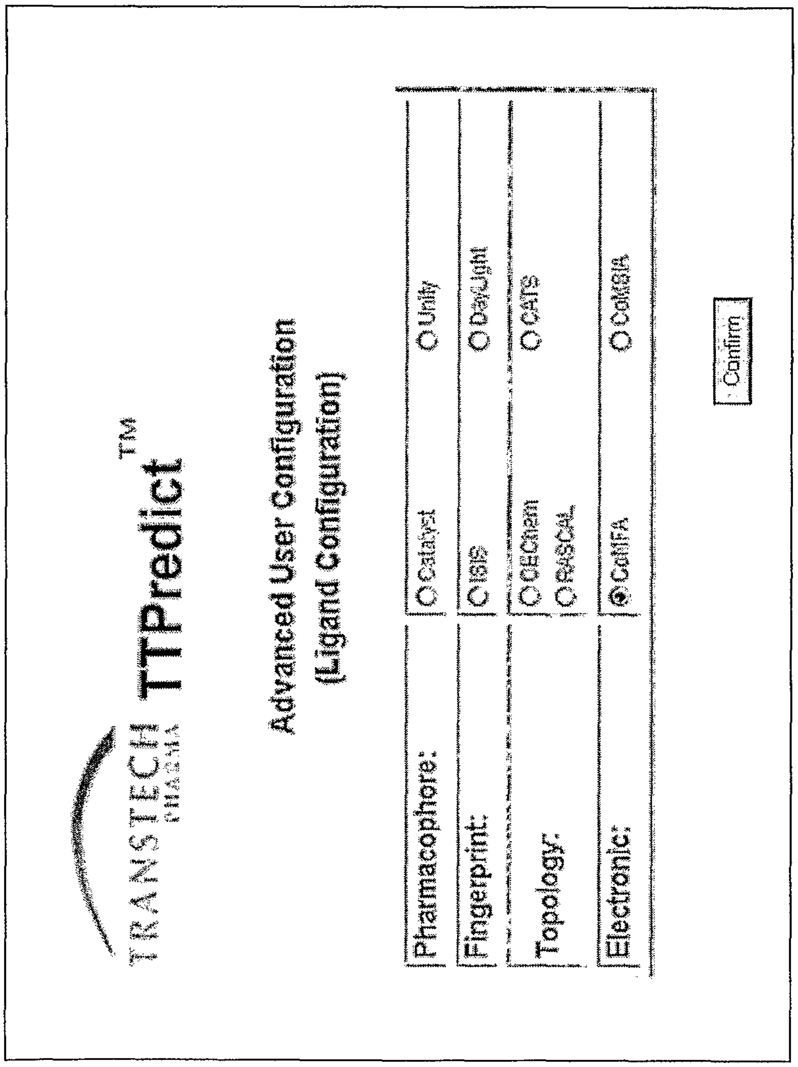

TRANSTECH
PHARMA
TTPredict TM
Advanced User Configuration
(Ligand Configuration)
Pharmacophore:
Catalyst
Unity
Fingerprint:
ISIS
DayLight
Topology:
OEChem
RASCAL
CATS
Electronic:
CoMFA
CoMSIA
Confirm

FIG. 21b

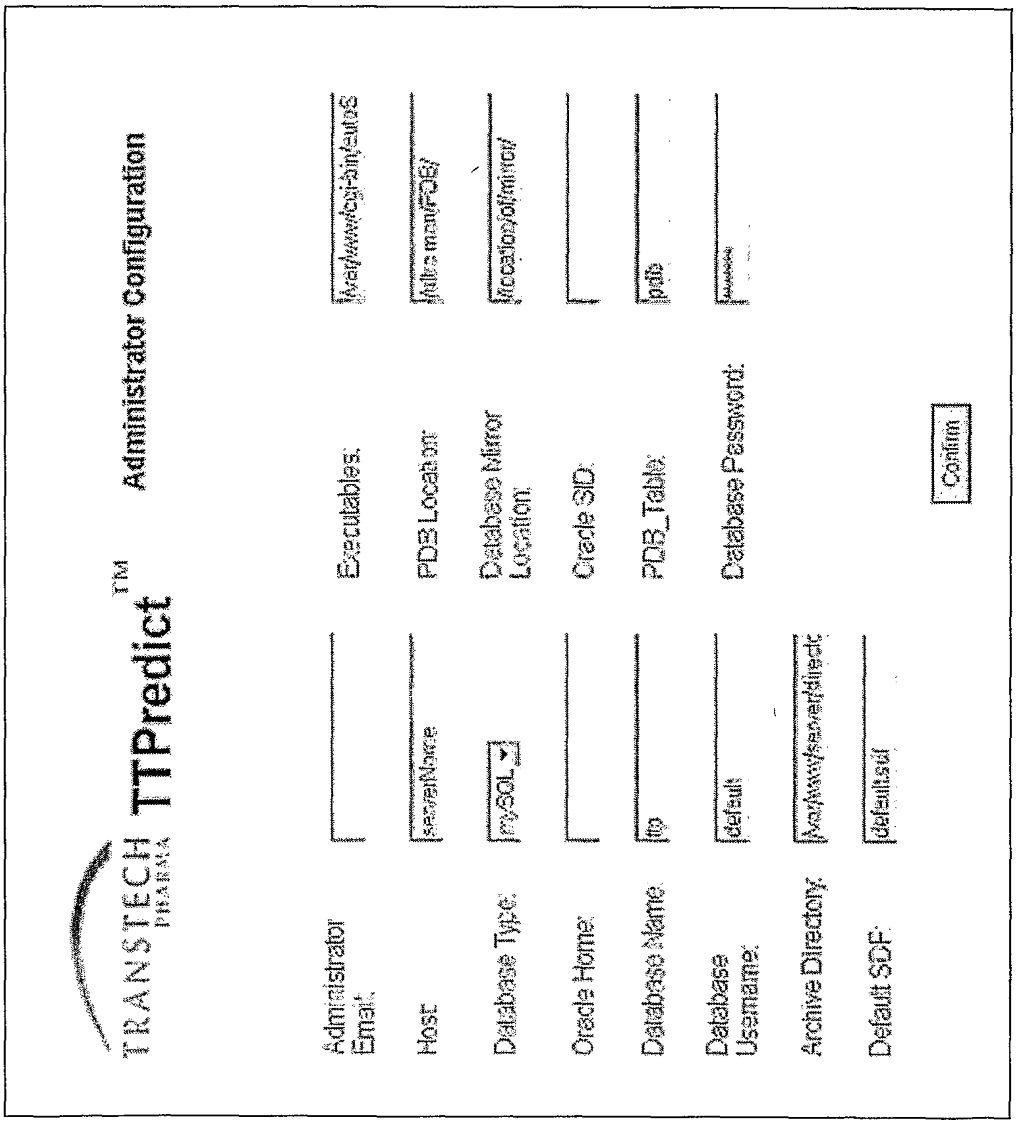

TRANSTECH PHARMA
TTPredict™
Administrator Configuration
Administrator Email:
Host:
serverName
Database Type:
mySQL
Oracle Home:
Database Name:
ttp
Database Username:
default
Archive Directory:
Default SDF:
defaultsdf
Executables:
PDB Location:
Database Mirror Location:
/location/of/mirror/
Oracle SID:
PDB_Table:
pdb
Database Password:
Confirm

FIG. 22

TRANSTECH PHARMA **TTPredict**™

TTPQ STATUS TABLE				
ID	STATUS	TYPE	WHO	WHERE
1		TTPostGene (search)	user1@ttpharma.com	server_6
33		TTPostGene (search)	user2@ttpharma.com	server_4
34		TTPostGene (search)	user3@ttpharma.com	server_6
35		TTPostGene (model creation)	user1@ttpharma.com	server_11
36		TTPSite	user1@ttpharma.com	server_3
37		TTPDock	user4@ttpharma.com	server_6
38		TTPDock	user1@ttpharma.com	server_6

LEGEND	
Icon	Definition
	Waiting
	Running
	DONE
ooo	Paused
	ERROR
	Undefined Type

FIG. 23

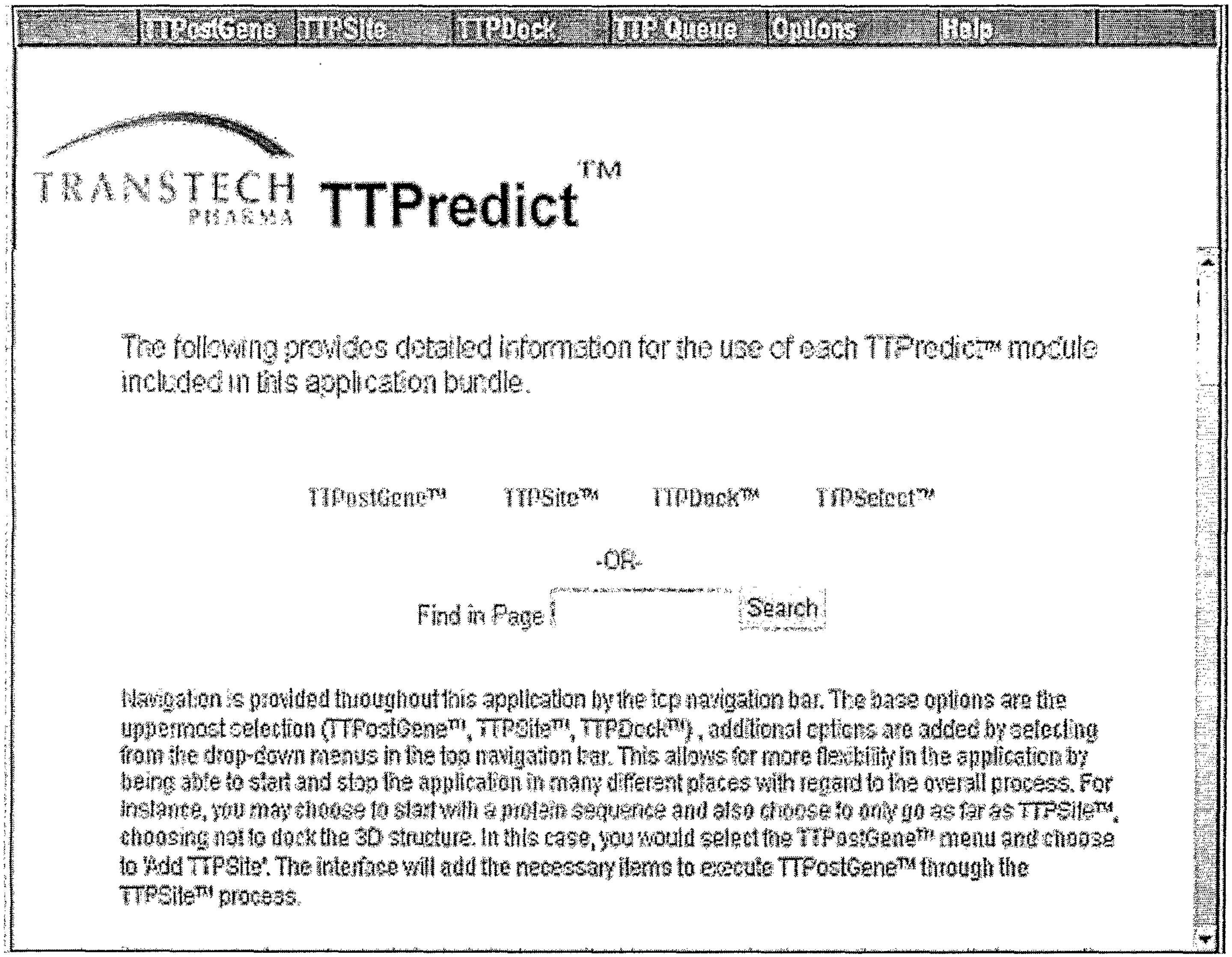

TTPostGene TTPSite TTPDock TTP Queue Options Help
TRANSTECH PHARMA
TTPredict™
The following provides detailed information for the use of each TTPredict™ module included in this application bundle.
TTPostGene™ TTPSite™ TTPDock™ TTPSelect™
-OR-
Find in Page
Search
Navigation is provided throughout this application by the top navigation bar. The base options are the uppermost selection (TTPostGene™, TTPSite™, TTPDock™), additional options are added by selecting from the drop-down menus in the top navigation bar. This allows for more flexibility in the application by being able to start and stop the application in many different places with regard to the overall process. For instance, you may choose to start with a protein sequence and also choose to only go as far as TTPSite™, choosing not to dock the 3D structure. In this case, you would select the TTPostGene™ menu and choose to 'Add TTPSite'. The interface will add the necessary items to execute TTPostGene™ through the TTPSite™ process.

FIG. 24

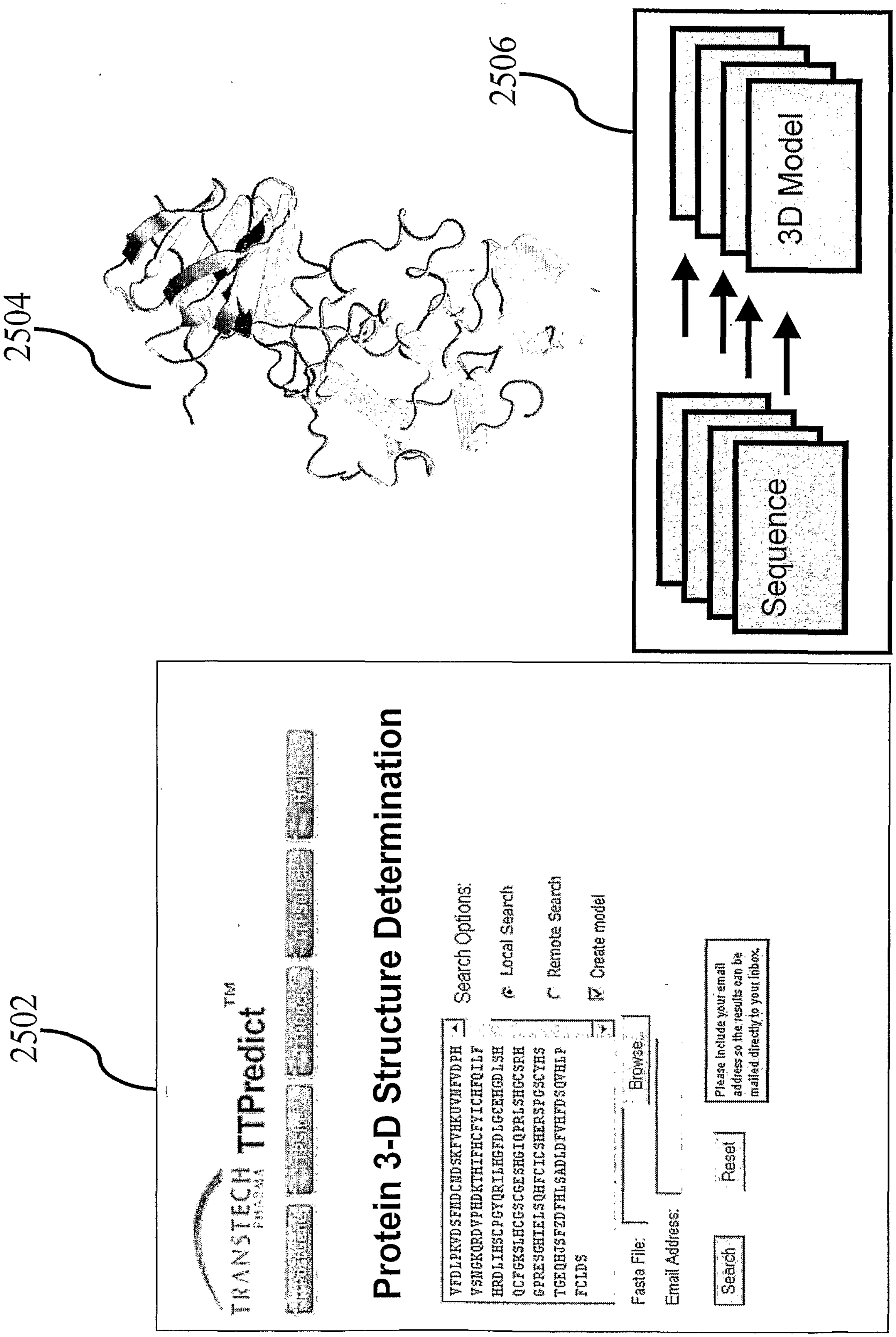

2502
2504
2506
TRANSTECH PHARMA
TTPredict™
Protein 3-D Structure Determination
Search Options:
Local Search
Remote Search
Create model
Fasta File:
Browse...
Email Address:
Search
Reset
Please include your email address so the results can be mailed directly to your inbox.
Sequence
3D Model

FIG. 25

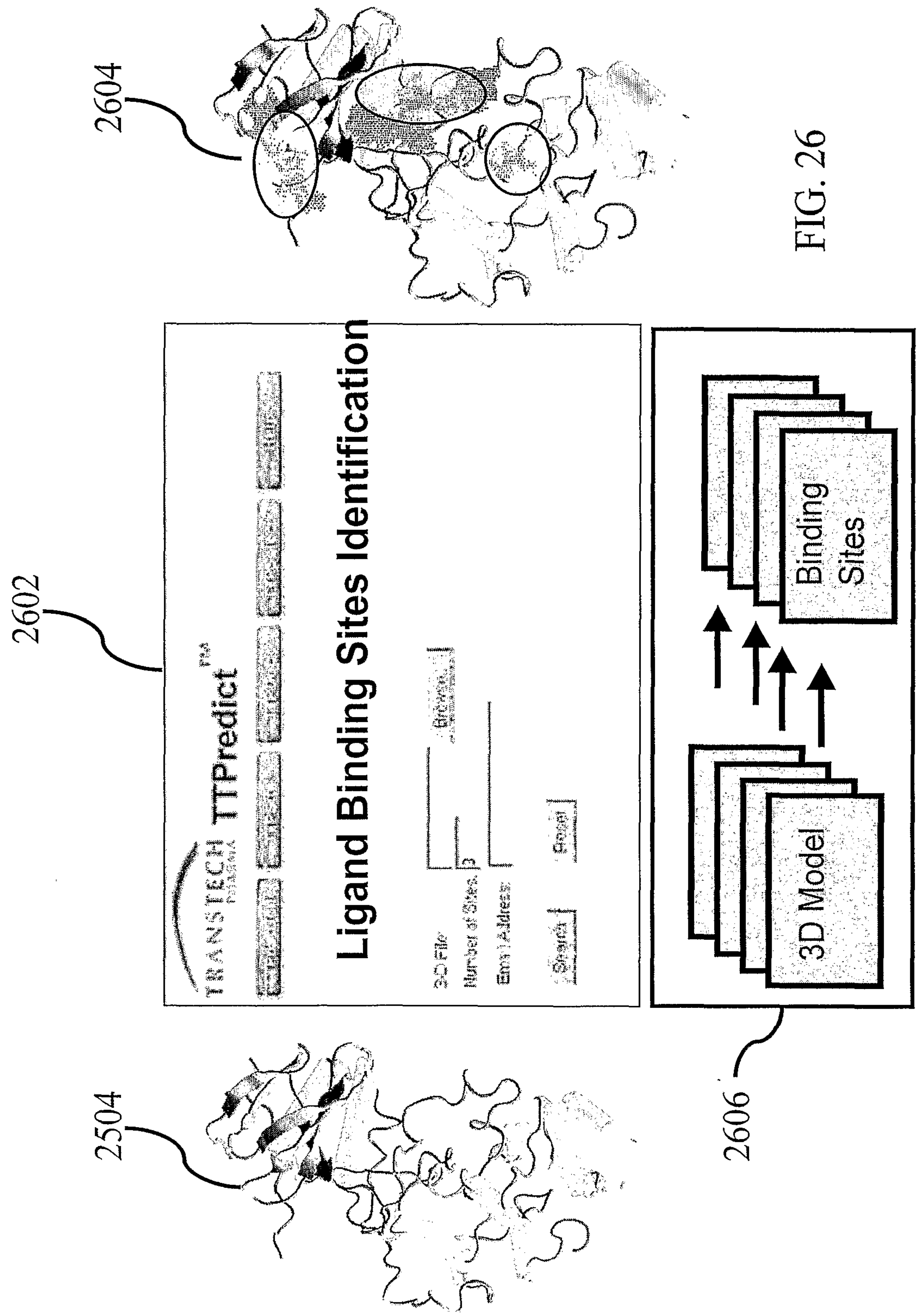

2604
2602
TRANSTECH
TTPredict™
Ligand Binding Sites Identification
3D File
Number of Sites
Email Address
Browse
Search
Reset
3D Model
Binding Sites
2504
2606

FIG. 26

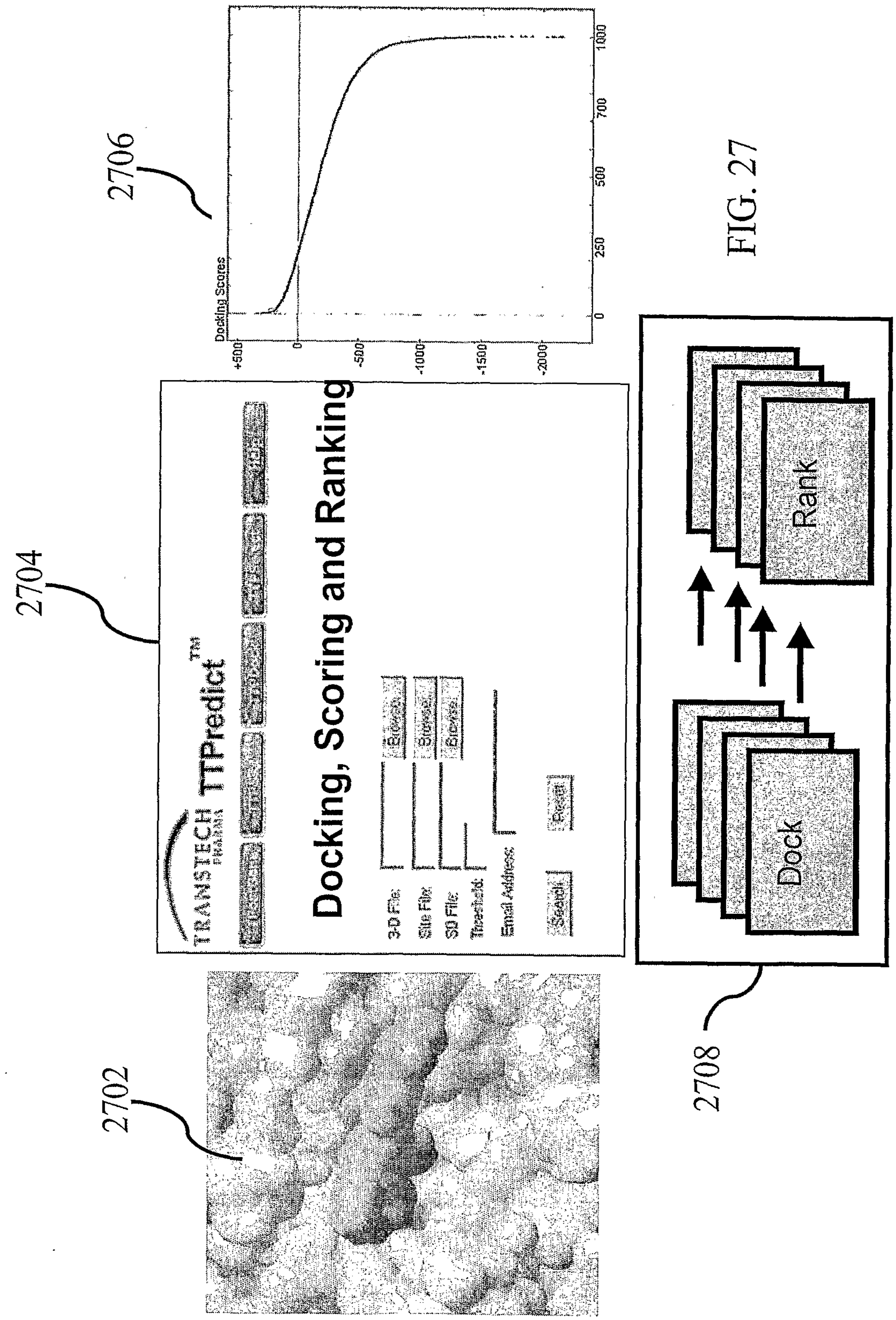

2702
2704
2706
2708
TRANSTECH PHARMA
TTPredict™
Docking, Scoring and Ranking
3-D File:
Site File:
SD File:
Threshold:
Email Address:
Browse
Browse
Browse
Search
Reset
Docking Scores
+500
0
-500
-1000
-1500
-2000
0
250
500
700
1000
Dock
Rank

FIG. 27